



## KERJA PRAKTIK - IF184801

Rancang Bangun Aplikasi Berbasis Web Sebagai  
Penyebaran Informasi Kejadian Secara *Real-Time*  
Menggunakan Kerangka Kerja Laravel

Laboratorium Komputasi Cerdas dan Visi, Departemen  
Teknik Informatika - ITS

Jl. Teknik Kimia - Gedung Departemen Teknik  
Informatika, Kampus Institut Teknologi Sepuluh  
Nopember Surabaya

Jalan Raya ITS, Sukolilo, Surabaya 60111

Periode: 1 Juni 2020 - 1 Oktober 2020

Oleh:

Sherly Rosa Anggraeni

05111740000018

Timothyus Tanner

05111740000103

Pembimbing Departemen

Dr.Eng. Chastine Fatichah, S.Kom, M.Kom

Pembimbing Lapangan

Dr.Eng. Chastine Fatichah, S.Kom, M.Kom

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2020



**KERJA PRAKTIK - IF184801**

**Rancang Bangun Aplikasi Berbasis Web Sebagai  
Penyebaran Informasi Kejadian Secara *Real-Time*  
Menggunakan Kerangka Kerja Laravel**

**Laboratorium Komputasi Cerdas dan Visi, Departemen Teknik  
Informatika - ITS**

**Jl. Teknik Kimia - Gedung Departemen Teknik Informatika,  
Kampus Institut Teknologi Sepuluh Nopember Surabaya  
Jalan Raya ITS, Sukolilo, Surabaya 60111**

**Periode: 1 Juni 2020 - 1 Oktober 2020**

Oleh:

Sherly Rosa Anggraeni 05111740000018

Timothyus Tanner 05111740000103

Pembimbing Departemen

Dr.Eng. Chastine Fatichah, S.Kom, M.Kom

Pembimbing Lapangan

Dr.Eng. Chastine Fatichah, S.Kom, M.Kom

**DEPARTEMEN TEKNIK INFORMATIKA**

**Fakultas Teknologi Elektro dan Informatika Cerdas**

**Institut Teknologi Sepuluh Nopember**

**Surabaya 2020**

*[Halaman ini sengaja dikosongkan]*

## LEMBAR PENGESAHAN KERJA PRAKTIK

### Rancang Bangun Aplikasi Berbasis Web Sebagai Penyebaran Informasi Kejadian Secara *Real-Time* Menggunakan Kerangka Kerja Laravel

Oleh:

**Sherly Rosa Anggraeni**  
**Timothyus Tanner**

05111740000018  
05111740000103

Disetujui oleh Pembimbing Kerja Praktik:

1. Dr.Eng. Chastine Fatichah, S.Kom, M.Kom  
NIP. 19751220 200112 2 002



(Pembimbing Departemen)

2. Dr.Eng. Chastine Fatichah, S.Kom, M.Kom  
NIP. 19751220 200112 2 002

(Pembimbing Lapangan)

# **Rancang Bangun Aplikasi Berbasis Web Sebagai Penyebaran Informasi Kejadian Secara *Real-Time* Menggunakan Kerangka Kerja Laravel**

Nama Mahasiswa : Sherly Rosa Anggraeni  
(0511174000018)

Timothyus Tanner  
(05111740000103)

Departemen : Teknik Informatika FTEIC-ITS

Pembimbing Departemen : Dr.Eng. Chastine Fatichah,  
S.Kom, M.Kom

Pembimbing Lapangan : Dr.Eng. Chastine Fatichah,  
S.Kom, M.Kom

## **ABSTRAK**

*Berkembang pesatnya dunia teknologi informasi memudahkan manusia untuk bertukar informasi dengan cepat dan mudah melalui media sosial. Salah satu media sosial yang sering digunakan adalah twitter. Contoh dari user influence yang menggunakan twitter sebagai sarana berbagai informasi kejadian seperti BMKG, Dishub Surabaya, dan Radio Suara Surabaya.*

*Kejadian penting seperti kecelakaan, kebakaran, kemacetan atau insiden lainnya yang dapat menimbulkan dampak negatif kepada masyarakat perlu diketahui sedini mungkin oleh masyarakat umum agar dapat terhindar dari dampak kejadian. Sistem ini dibuat untuk memfasilitasi masyarakat dan pihak berwenang dalam penyebaran informasi terkait kejadian penting secara daring.*

***Kata kunci: Aplikasi Web, Incident Report, Informasi Kejadian, Media Sosial***

*[Halaman ini sengaja dikosongkan]*

## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena berkat rahmat dan lindungan-Nya penulis dapat melaksanakan salah satu kewajiban kami sebagai mahasiswa Departemen Teknik Informatika, yakni Kerja Praktik (KP).

Penulis menyadari masih ada kekurangan baik dalam pelaksanaan KP maupun penyusunan buku laporan ini. Namun, penulis berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi. Penulis mengharapkan kritik dan saran yang membangun untuk kesempurnaan buku laporan KP ini.

Melalui buku ini, penulis juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu, baik langsung maupun tidak langsung, dalam pelaksanaan KP hingga penyusunan laporan. Orang-orang tersebut antara lain adalah:

1. Kedua orang tua penulis.
2. Ibu Dr.Eng. Chastine Fatichah, S.Kom, M.Kom selaku dosen pembimbing Kerja Praktik.
3. Bapak Ibu dosen ITS yang ikut terlibat dalam simulasi penggunaan aplikasi sebelum peluncuran aplikasi dilaksanakan.
4. Teman-teman penulis yang senantiasa memberikan semangat ketika penulis melaksanakan KP.

Surabaya, September 2020  
Penulis

*[Halaman ini sengaja dikosongkan]*



# DAFTAR ISI

LEMBAR PENGESAHAN .....	iv
ABSTRAK.....	v
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR TABEL .....	xiii
DAFTAR GAMBAR.....	xiv
BAB I.....	1
1.1.        Latar Belakang .....	1
1.2.        Tujuan.....	1
1.3.        Manfaat.....	2
1.4.        Rumusan Masalah .....	2
1.5.        Lokasi dan Waktu Kerja Praktik .....	3
1.6.        Metodologi Kerja Praktik .....	3
1.7.        Sistematika Laporan .....	4
BAB II.....	6
1.1.        Sejarah Instansi.....	6
1.2.        Visi dan Misi Instansi.....	6
1.3.        Laboratorium Komputasi Cerdas dan Visi .....	7
BAB III.....	10
3.1.        Pemrograman Web .....	10
3.2.        Javascript.....	10
3.3.        PHP.....	10
3.4.        Bootstrap CSS .....	10
3.5.        JQuery .....	11
3.6.        Python.....	11

3.7.	Laravel.....	11
3.8.	MySQL.....	11
3.9.	VSCode .....	11
3.10.	Deep Learning .....	12
3.11.	Ekstraksi Informasi.....	12
3.12.	Neuro Named Entity Recognition ( <i>NeuroNER</i> ) .....	12
3.13.	Neural Relation Extraction ( <i>NeuralRE</i> ) .....	13
3.14.	RCNN Classification .....	13
3.15.	Fast Approximate String Matching .....	13
3.16.	Directed Graph .....	13
BAB IV.....		14
4.1.	Analisis Sistem.....	14
4.2.	Diagram Kasus Penggunaan.....	16
4.3.	Spesifikasi Kasus Penggunaan .....	17
4.4.	Diagram Aktivitas .....	20
BAB V.....		22
5.1.	Implementasi Pengambilan Data.....	22
5.2.	Implementasi Web.....	36
BAB VI.....		52
6.1.	Tujuan Pengujian.....	52
6.2.	Kriteria Pengujian.....	52
6.3.	Lingkungan Pengujian.....	52
6.4.	Skenario Pengujian.....	53
6.5.	Evaluasi Pengujian .....	53
6.6.	Tampilan Aplikasi .....	55
BAB VII.....		60
7.1.	Kesimpulan.....	60
7.2.	Saran.....	60

DAFTAR PUSTAKA.....61

BIODATA PENULIS.....64

*[Halaman ini sengaja dikosongkan]*

## DAFTAR TABEL

Tabel 4.1 Kebutuhan Fungsional .....	15
Tabel 4.2 Kebutuhan Non-Fungsional.....	16
Tabel 4.3 Tabel Use Case Melihat Peta Persebaran Kejadian Penting.....	17
Tabel 4.4 Tabel Use Case Melihat Data Tweet Kejadian Penting.....	18
Tabel 4.5 Tabel Use Case Melihat Rekapitulasi Kejadian .....	19
Tabel 6.1 Hasil Evaluasi Pengujian Aplikasi Incident Report.....	54

## DAFTAR GAMBAR

Gambar 4.1 Diagram Alur Sistem Incident Report .....	14
Gambar 4.2 Diagram Use Case Incident Report.....	17
Gambar 4.3 Diagram Aktivitas Peta Persebaran Kejadian Penting .....	20
Gambar 4.4 Diagram Aktivitas Data Tweet Kejadian Penting .....	21
Gambar 4.5 Diagram Aktivitas Menampilkan Rekapitulasi Kejadian .....	21
Gambar 6.1 Tampilan Lokasi Kejadian Penting .....	55
Gambar 6.2 Tampilan Pilihan Rentang Waktu Kejadian Penting .....	56
Gambar 6.3 Tampilan Data Tweet Kejadian Penting .....	57
Gambar 6.4 Tampilan Rekapitulasi Kejadian Penting Bulan .....	58
Gambar 6.5 Tampilan Rekapitulasi Kejadian Penting Kecamatan.....	59

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Berkembang pesatnya dunia teknologi informasi memudahkan manusia untuk bertukar informasi dengan cepat dan mudah melalui media sosial. Salah satu media sosial yang sering digunakan adalah *twitter*. Banyak pengguna memanfaatkan *twitter* sebagai sarana berbagi informasi kejadian penting, baik pengguna individu maupun pengguna sebagai representatif lembaga atau perusahaan yang disebut sebagai *User Influence* (UI). Contoh dari UI yang menggunakan *twitter* sebagai sarana berbagai informasi kejadian seperti BMKG, Dishub Surabaya, dan Radio Suara Surabaya.

Kejadian penting seperti kecelakaan, kebakaran, kemacetan atau insiden lainnya yang dapat menimbulkan dampak negatif kepada masyarakat perlu diketahui sedini mungkin oleh masyarakat umum agar dapat terhindar dari dampak kejadian. Dan untuk pihak berwenang, mengetahui informasi kejadian penting sedini mungkin merupakan suatu keharusan agar dapat segera melakukan tindakan untuk mengatasi masalah tersebut.

Sistem ini dibuat untuk memfasilitasi masyarakat dan pihak berwenang dalam penyebaran informasi terkait kejadian penting secara daring. Aplikasi menyediakan fitur berbentuk *maps* yang menampilkan persebaran titik beberapa kategori kejadian secara *real time* dan rekapitulasi kejadian pada kategori tertentu dalam bentuk grafik.

### **1.2. Tujuan**

Tujuan Kerja Praktik ini adalah untuk menyelesaikan kewajiban kuliah kerja praktik di Institut Teknologi Sepuluh Nopember dengan beban 2 SKS. Selain itu juga untuk memenuhi kebutuhan yang diperlukan oleh penelitian laboratorium Komputasi Cerdas dan Visi dengan mengimplementasikan aplikasi berbasis web penyebaran informasi kejadian perkotaan di wilayah Surabaya dan sekitarnya. Aplikasi ini menitikberatkan pada fungsionalitas user.

Kerja praktik ini memiliki beberapa tujuan, yaitu membuat program yang dapat:

1. Mengambil data kejadian pada *twitter* secara *real time*.
2. Mengklasifikasikan data *twitter* menjadi kategori kejadian (bencana alam, lalu-lintas, kebakaran, lain-lain).
3. Menyajikan dalam bentuk grafik rekapitulasi kejadian penting berdasarkan bulan dan kecamatan yang di filter berdasarkan tahun.
4. Menyajikan lokasi kejadian penting secara visual melalui peta.

### **1.3. Manfaat**

Manfaat yang kami peroleh dalam pelaksanaan kerja praktik ini adalah:

1. Pengalaman dalam lingkungan kerja yang sebenarnya.
2. Meningkatkan kemampuan kerjasama dalam tim.

Manfaat yang dapat diperoleh dengan adanya Aplikasi *Incident Report* ini adalah:

1. Memudahkan masyarakat dalam mendapatkan informasi mengenai kejadian penting secara *real time*.

### **1.4. Rumusan Masalah**

Berikut ini rumusan masalah dalam pelaksanaan kerja praktik pembuatan Aplikasi *Incident Report*:

1. Bagaimana solusi efektif dalam pengambilan data kejadian pada *twitter* secara *real time*?
2. Bagaimana teknis pengklasifikasian data *twitter* menjadi kategori kejadian (bencana alam, lalu-lintas, kebakaran, lain-lain)?
3. Bagaimana penyajian rekapitulasi kejadian penting berdasarkan bulan dan kecamatan yang di filter berdasarkan tahun dalam bentuk grafik?
4. Bagaimana penyajian lokasi kejadian penting yang divisualisasikan melalui peta?



### **1.5. Lokasi dan Waktu Kerja Praktik**

Kerja praktik ini dilaksanakan pada waktu dan tempat sebagai berikut:

Lokasi : Work From Home (di rumah masing-masing)  
Waktu : 1 Juni 2020 – 1 Oktober 2020  
Hari Kerja : Senin - Jumat  
Jam Kerja : 08.00 WIB - 17.00 WIB

### **1.6. Metodologi Kerja Praktik**

Tahapan pengerjaan kerja praktik dapat dijabarkan sebagai berikut:

#### **1. Perumusan Masalah**

Untuk mengetahui domain dan fungsionalitas, dijelaskan secara rinci bagaimana sistem yang harus dibuat. Penjelasan oleh Pembimbing Lapangan kerja praktik menghasilkan beberapa catatan mengenai gambaran secara garis besar tentang kebutuhan atau fitur apa saja yang harus ada dalam sistem. Setelah mendapatkan gambaran sistem, terdapat diskusi lebih lanjut dengan diputuskan bahwa aplikasi menggunakan bahasa pemrograman PHP dan Javascript serta menggunakan database MySQL Server dengan menerapkan kerangka kerja Laravel. Selanjutnya, programmer dapat menggunakan Bahasa pemrograman lainnya sebagai pendukung pelengkapan fitur yang dibuat.

#### **2. Studi Literatur**

Setelah ditentukan rancangan database, bahasa pemrograman, serta tools tambahan yang akan digunakan, dilakukan studi literatur mengenai cara implementasinya dalam membangun sistem sesuai yang dibutuhkan. Pada tahap ini dilakukan proses pencarian, pembelajaran, pengumpulan dan pemahaman informasi serta literatur yang berkaitan untuk membantu dalam implementasi aplikasi ini. Informasi bisa didapat dari internet untuk istilah-istilah umum yang digunakan dalam pengimplementasian suatu sistem informasi.

### **3. Analisis dan Perancangan Sistem**

Langkah ini meliputi penjelasan awal tentang sistem. Bagaimana cara kerja sistem dengan skenario tertentu. Dari penjelasan awal telah didapatkan beberapa kebutuhan fungsional dan non-fungsional secara garis besar. Kemudian dilanjutkan dengan memperjelas spesifikasi kebutuhan-kebutuhan tersebut. Dibuatlah sebuah diagram kasus penggunaan yang mewakili skenario-skenario untuk penggunaan sistem aplikasi. Dilanjutkan dengan diskusi bersama pembimbing lapangan untuk mengetahui kebutuhan-kebutuhan tersebut telah tepat atau tidak.

### **4. Implementasi**

Implementasi sistem didasarkan pada perancangan dan analisis sebelumnya. Penentuan atribut dan atau fitur, tipe data, dan format luaran yang akan dibuat pada model juga didasarkan pada analisis sebelumnya sesuai dengan kebutuhan.

Pengerjaan dilakukan dengan pelaporan setiap minggu yang dipantau oleh dosen pembimbing dan pembimbing lapangan dengan adanya meeting terencana minimal satu kali dalam satu minggu. Selain itu, pembimbing lapangan dan dosen pembimbing juga memberikan masukan-masukan apabila terdapat kendala.

### **5. Pengujian dan Evaluasi**

Pengujian dilakukan dengan menguji fitur-fitur yang telah dibuat. Kesesuaian sistem dengan kebutuhan akan menentukan keberhasilan dalam pengujian. Hal ini akan menghasilkan hasil evaluasi apakah sistem sudah sesuai dengan tujuan dan kebutuhan atau belum.

## **1.7. Sistematika Laporan**

Laporan KP ini terdiri dari tujuh bab dengan rincian sebagai berikut:

### **1. Bab I Pendahuluan**

Pada bab ini dijelaskan tentang latar belakang permasalahan, tujuan, waktu pelaksanaan, serta sistematika

pengerjaan KP dan juga penulisan laporan KP.

**2. Bab II Profil Perusahaan**

Pada bab ini, dijelaskan secara rinci tentang profil Laboratorium Komputasi Cerdas dan Visi Departemen Teknik Informatika ITS.

**3. Bab III Tinjauan Pustaka**

Pada bab ini, dijelaskan mengenai konsep-konsep pembuatan model, teknologi, tinjauan pustaka dan literatur yang digunakan dalam penyelesaian KP.

**4. Bab IV Analisis dan Perancangan Sistem**

Pada bab ini, dijelaskan hasil pembelajaran atau analisis terhadap apa saja yang diperlukan dan harus diperhatikan dalam pengembangan aplikasi yang dikerjakan selama KP.

**5. Bab V Implementasi Sistem**

Pada bab ini, berisi penjelasan tahap-tahap yang dilakukan untuk proses implementasi aplikasi.

**6. Bab VI Pengujian dan Evaluasi**

Pada bab ini, dijelaskan tentang hasil pengujian dan evaluasi dari sistem yang telah dikembangkan selama pelaksanaan KP.

**7. Bab VII Kesimpulan dan Saran**

Pada bab ini, dipaparkan kesimpulan yang dapat diambil dan juga saran selama pengerjaan KP.

## **BAB II**

### **PROFIL INSTANSI**

#### **1.1. Sejarah Instansi**

Pendidikan tinggi diarahkan untuk mempersiapkan bangsa Indonesia dalam menghadapi era pembangunan industri dan informasi. Untuk itu pemerintah melalui Direktorat Jendral Pendidikan Tinggi pada tahun 1985 menginstruksikan untuk membuka Program Studi S1 baru untuk bidang ilmu teknologi komputer di empat universitas atau institut di mana ITS termasuk di dalamnya. Di ITS, program ini awalnya diberi nama Program Studi Teknik Komputer. Namun sejak tahun 1993, nama Program Studi Teknik Komputer diubah menjadi Jurusan Teknik Komputer. Akhirnya, pada tahun 1996 secara resmi jurusan ini berganti nama menjadi Jurusan Teknik Informatika berdasarkan Surat Keputusan Direktur Jendral Pendidikan Tinggi Nomor 224/DIKTI/Kep/1996, tanggal 11 Juli 1996. Pada saat ini, Departemen Teknik Informatika memperoleh nilai akreditasi A berdasarkan Surat Keputusan Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT) Nomor 003/BAN-PT/Ak-X/S1/V/2006, tanggal 18 Mei 2006 [1].

#### **1.2. Visi dan Misi Instansi**

##### **Visi Departemen Teknik Informatika**

Visi Departemen Teknik Informatika adalah menjadi inovator bidang informatika yang unggul di tingkat nasional dengan reputasi internasional, serta berperan aktif dalam upaya memajukan dan mensejahterakan bangsa

##### **Misi Departemen Teknik Informatika**

1. Menyelenggarakan proses pembelajaran yang berkualitas, dan memenuhi standar nasional maupun internasional.
2. Melaksanakan penelitian yang inovatif, bermutu, dan bermanfaat.
3. Meningkatkan pemanfaatan teknologi informasi dan komunikasi untuk masyarakat.
4. Menjalinkan kemitraan dengan berbagai lembaga, baik di dalam maupun di luar negeri.

### **1.3. Laboratorium Komputasi Cerdas dan Visi**

#### **1.3.1. Pendahuluan**

Departemen Teknik Informatika memiliki delapan laboratorium bidang riset guna menunjang pendidikan mahasiswa yaitu Laboratorium Rekayasa Perangkat Lunak (RPL), Laboratorium Komputasi Berbasis Jaringan (KBJ), Laboratorium Komputasi Cerdas Visi (KCV), Laboratorium Arsitektur dan Jaringan Komputer (AJK), Laboratorium Grafika, Interaksi dan Game (GIGa), Laboratorium Algoritma Pemrograman (AP), Laboratorium Manajemen Cerdas Informasi (MCI), dan Laboratorium Pemodelan dan Komputasi Terapan (PKT). Laboratorium Komputasi Cerdas dan Visi berfokus pada kemampuan lulusan dalam memanipulasi dan menganalisis data citra pada berbagai bidang aplikasi (a.l. biomedika, industri), kemampuan menerapkan metode sistem cerdas pada berbagai bidang aplikasi dan kemampuan memodelkan dan mengoptimasikan sistem nyata [2].

#### **1.3.2. Mata Kuliah**

Yang diampu oleh Rumpun Mata Kuliah KCV, antara lain:

1. Pengolahan Citra Digital,
2. Data Mining,
3. Komputasi Biomedik,
4. Visi Komputer,
5. Sistem Temu Kembali Informasi,
6. Robotika,
7. Analisis Jejaring Sosial,
8. *Deep Learning*.

### **1.3.3. Struktur Organisasi Laboratorium Komputasi Cerdas dan Visi**

1. Dr.Eng. Nanik Suciati, S.Kom,M.Kom sebagai Ketua RMK
2. Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.
3. Prof.Ir. Handayani Tjandrasa, M.Sc Ph.D
4. Prof. Dr. Agus Zainal Arifin, S.Kom., M.Kom.
5. Dini Adni Navastara, S.Kom, M.Sc.

### **1.3.4. Penelitian**

1. Pengembangan Perangkat Lunak Sistem Pengenal Keadaan Mental Manusia Berdasarkan Gelombang Aktivitas Otak EEG, 2017.
2. Pengembangan Sistem Pengukuran Otomatis Kualitas Tulang Berdasarkan Tulang Trabekula pada Leher Kondilus, 2017.
3. Deteksi Awal Jenis Pohon Mangga Dengan Aplikasi Perangkat Mobil eAndroid Berdasarkan Warna dan Tekstur Daun, 2017.
4. Deep Learning Untuk Temu Kembali Data Citra Berjumlah Besar, 2017.
5. Kombinasi Informasi Spasial dan Fitur Garis pada Adaptive Classifier Untuk Segmentasi Tulang Trabekula Pada Dental Panoramic Radiograph, 2017.
6. Pengenalan Wajah pada Data Video untuk Pengawasan Ruang, 2018.
7. Klasterisasi Tren Topik Penelitian Untuk Rekomendasi Pengadaan Bahan Pustaka Di Perpustakaan Its, 2018.
8. Sistem Cerdas Untuk Temu Kembali Informasi Pada Dokumen Berbahasa Arab, 2018.
9. Sistem Deteksi Dini Osteoporosis Menggunakan Fitur Kondilus pada Citra Panorama Gigi, 2018.
10. Deteksi Kejadian di Perkotaan melalui Crowdsourcing Berbasis Informasi Spasial dan Temporal menggunakan Metode Deep Learning, 2018.
11. Educational Data Mining Berbasis Klasifikasi Untuk Menganalisis Mahasiswa Berpotensi Berhenti Kuliah

- Pada Kasus Imbalanced Class Distribution, 2018.
12. Deteksi Awal Jenis Pohon Mangga Dengan Aplikasi Perangkat Mobile Android Berdasarkan Warna dan Tekstur Daun, 2018.
  13. Pembuatan Sistem Penerjemah Bahasa Isyarat Bahasa Indonesia untuk Alat Bantu Komunikasi bagi Tuna Rungu Menggunakan Deep Learning, 2018.
  14. Kombinasi Fitur Histogram of Oriented Gradients dan Local Binary Patterns untuk Klasifikasi Jenis Kendaraan, 2018.
  15. Klasifikasi jenis kejadian menggunakan kombinasi NeuroNER dan Recurrent Convolutional Neural Network pada data Twitter, 2018.

## **BAB III**

### **TINJAUAN PUSTAKA**

Pada bab ini, akan dijelaskan mengenai dasar teori yang digunakan selama proses pengerjaan dan pengembangan aplikasi.

#### **3.1. Pemrograman Web**

Aplikasi web adalah sebuah sistem informasi yang mendukung interaksi pengguna melalui antarmuka berbasis Web dan diakses melalui jaringan komputer yang sesuai [3]. Pemrograman web adalah pemrograman yang digunakan untuk membangun aplikasi web tersebut.

#### **3.2. Javascript**

JavaScript adalah bahasa pemrograman web yang bersifat Client Side Programming Language yang digunakan untuk menambahkan fitur interaktif [4].

#### **3.3. PHP**

PHP adalah bahasa pemrograman yang berbentuk script pengelola web service yang menggunakan protokol HTTP dan diletakkan didalam web server. Secara singkatnya, PHP ini digunakan untuk menghubungkan database dengan javascript aplikasi tersebut [5].

#### **3.4. Bootstrap CSS**

Bootstrap adalah sebuah library framework CSS yang di buat khusus untuk bagian pengembangan front-end website yang dikembangkan oleh tim twitter. Bootstrap merupakan salah satu framework HTML, CSS dan javascript yang paling populer di kalangan web developer dikarenakan website akan menjadi lebih fleksibel, nyaman dan cepat [6].



### **3.5. JQuery**

JQuery merupakan sebuah library JavaScript multiplatform yang dirancang untuk memudahkan penyusunan client-side script pada file HTML. JQuery juga dapat diartikan sebagai kumpulan fungsi-fungsi JavaScript yang memudahkan penulisan kode JavaScript dalam pembuatan sebuah aplikasi [7].

### **3.6. Python**

Python adalah bahasa pemrograman interpretatif multiguna. Python sendiri merupakan bahasa Pemrograman yang telah populer serta banyak digunakan di Data Analysis, para Software Engineers, dan Data Scientists dalam mengoperasikan pembangunan di salah satu aplikasi serta menggali secara lebih mendalam pada machine learning [8].

### **3.7. Laravel**

Laravel adalah sebuah MVC web development framework yang didesain untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan dan perbaikan. Selain itu, Laravel juga dipercaya dalam meningkatkan produktifitas pekerjaan dengan syntax yang bersih [9].

### **3.8. MySQL**

MySQL adalah suatu perangkat lunak database relasi yang didistribusikan gratis dibawah lisensi GPL (General Public License). Setiap orang bebas menggunakan MySQL namun tidak diperbolehkan untuk menjadikannya sebagai kepentingan komersial. MySQL digunakan sebagai tools untuk membuat database, mencari, menghapus, menambah, impor dan ekspor table dari dan atau ke file CSV [10].

### **3.9. VSCode**

Visual Studio Code atau VSCode adalah sebuah text editor yang digunakan untuk menulis kode program [11]. VSCode telah terintegrasi dengan Emmet, sehingga dapat memudahkan penulisan

kode menjadi lebih cepat dan efisien. Juga terdapat fitur Git yang memudahkan untuk bekerja dengan tim.

### **3.10. Deep Learning**

Deep Learning adalah salah satu jenis algoritma jaringan saraf tiruan yang menggunakan metadata sebagai input dan mengolahnya menggunakan sejumlah lapisan tersembunyi (hidden layer) transformasi non linier dari data masukan untuk menghitung nilai output. Algoritma pada Deep Learning memiliki fitur yang unik yaitu sebuah fitur yang mampu mengekstraksi secara otomatis. Hal ini berarti algoritma yang dimilikinya secara otomatis dapat menangkap fitur yang relevan sebagai keperluan dalam pemecahan suatu masalah. Algoritma semacam ini sangat penting dalam sebuah kecerdasan buatan karena mampu mengurangi beban pemrograman dalam memilih fitur yang eksplisit. Dan, algoritma ini dapat digunakan untuk memecahkan permasalahan yang perlu pengawasan (supervised), tanpa pengawasan (unsupervised), dan semi terawasi (semi supervised) [12].

### **3.11. Ekstraksi Informasi**

Ekstraksi informasi (IE) adalah tugas mengekstraksi informasi terstruktur secara otomatis dari dokumen yang tidak terstruktur dan / atau semi-terstruktur yang dapat dibaca mesin dan sumber lain yang diwakili secara elektronik. Dalam sebagian besar kasus, aktivitas ini berkaitan dengan pemrosesan teks bahasa manusia dengan menggunakan natural language processing (NLP). Aktivitas terkini dalam pemrosesan dokumen multimedia seperti anotasi otomatis dan ekstraksi konten dari gambar / audio / video / dokumen dapat dilihat sebagai ekstraksi informasi [13].

### **3.12. Neuro Named Entity Recognition (*NeuroNER*)**

*NeuroNER* adalah program yang melakukan NER. *NeuroNER* menyajikan keunggulan berikut dibandingkan sistem NER yang ada, yaitu memanfaatkan kemampuan prediksi mutakhir dari jaringan saraf (*deep learning*), memungkinkan pengguna membuat atau mengubah anotasi untuk korpus baru atau yang sudah ada, mengetahui lintas platform, open source, tersedia secara gratis, dan

langsung digunakan [14].

### 3.13. Neural Relation Extraction (*NeuralRE*)

*NeuralRE* adalah metode yang bertujuan melakukan ekstraksi informasi berupa relasi yang digunakan untuk mengetahui hubungan antar entitas pada teks kalimat [15].

### 3.14. Recurrent Convolutional Neural Network (RCNN) Classification

*Recurrent Convolutional Neural Network* (RCNN) merupakan algoritma pengembangan dari RNN dan CNN. Algoritma klasifikasi RCNN menggunakan *bi-directional recurrent* struktur yang diharapkan dapat mengenalkan sangat sedikit *noise* fitur dibandingkan model *neural network* tradisional berbasis *window* [16].

### 3.15. Fast Approximate String Matching

*Fast Approximate String Matching* (FASM) dengan memanfaatkan metode *edit distance Damerau-Levenshtein Distance* merupakan modifikasi dari teknik pencarian lainnya yaitu *fuzzy string matching* (FSM) yang memanfaatkan teknik *edit distance Levenshtein Distance* untuk mencari string yang cocok dengan prediksi kemungkinan yang paling mendekati (bukan tepatnya atau pasti) [17].

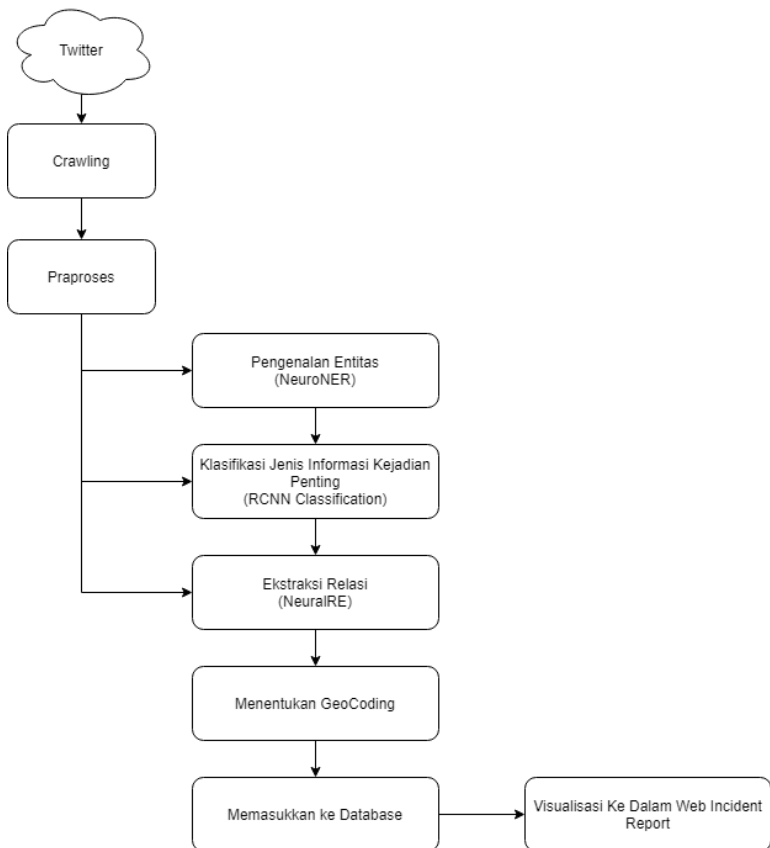
### 3.16. Directed Graph

Graf atau jaringan adalah kumpulan node (V) bersama yang dihubungkan dengan *edges* (E) untuk menghubungkan antar node. Graph Berarah (*directedgraph/digraph*), adalah graph yang simpulnya tidak hanya menghubungkan 2 vertex tetapi juga menunjukkan arah [18].

## BAB IV ANALISIS DAN PERANCANGAN SISTEM

### 4.1. Analisis Sistem

Pada bab ini akan dijelaskan mengenai tahapan dalam membangun Aplikasi Web *Incident Report* berupa analisis dari sistem yang digambarkan pada gambar 4.1.



*Gambar 4.1 Diagram Alur Sistem Incident Report*

#### 4.1.1. Definisi Umum Aplikasi

Secara umum, Aplikasi *Incident Report* merupakan sistem berbasis web yang digunakan untuk menyampaikan informasi mengenai 4 jenis kejadian yaitu Lalu Lintas, Kebakaran, Bencana Alam, dan Lain-Lain di daerah Jawa Timur. Aplikasi ini diharapkan dapat mempermudah masyarakat Jawa Timur dalam mengakses informasi kejadian penting. Adapun pengguna aplikasi ini didefinisikan sebagai berikut :

- Masyarakat Jawa Timur yang dapat melihat informasi kejadian penting.

#### 4.1.2. Analisis Kebutuhan

Dalam aplikasi ini, terdapat fungsi-fungsi yang harus dipenuhi oleh sistem. Kebutuhan ini terbagi ke dalam dua jenis, yakni kebutuhan fungsional dan kebutuhan non-fungsional.

##### 4.1.2.1. Kebutuhan Fungsional

Kebutuhan fungsional pada aplikasi ini menjelaskan bagaimana sistem ini bekerja. Kebutuhan fungsional dari aplikasi *Incident Report* dijelaskan pada Tabel 4.1.

*Tabel 4.1 Kebutuhan Fungsional*

Kode Kebutuhan	Deskripsi Kebutuhan
F01	Melihat peta persebaran kejadian penting
F02	Melihat data <i>tweet</i> kejadian penting
F03	Melihat rekapitulasi kejadian

#### 4.1.2.2. Kebutuhan Non-Fungsional

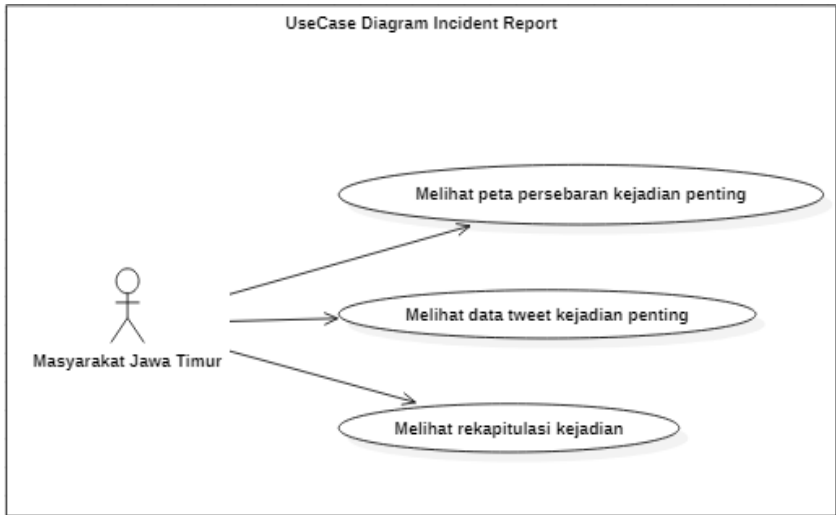
Kebutuhan non-fungsional adalah kebutuhan pengguna untuk mendefinisikan pengguna untuk mendefinisikan bagaimana batasan dan karakteristik dari sebuah sistem yang dibangun. Kebutuhan non-fungsional dari aplikasi *Incident Report* terdapat pada Tabel 4.2.

*Tabel 4.2 Kebutuhan Non-Fungsional*

<b>Kode Kebutuhan</b>	<b>Deskripsi Kebutuhan</b>	<b>Kualitas</b>
NF01	Sistem dapat diakses 24 jam	Portability
NF02	Sistem dibuat dalam bentuk aplikasi berbasis web	Portability

#### 4.2. Diagram Kasus Penggunaan

Daftar kebutuhan fungsional dapat dipresentasikan menjadi diagram kasus penggunaan (Use Case Diagram) sehingga memudahkan untuk dipahami. Use Case Diagram yang telah dibuat dapat dilihat pada Gambar 4.2.



*Gambar 4.2 Diagram Use Case Incident Report*

### 4.3. Spesifikasi Kasus Penggunaan

#### 4.3.1. Melihat Peta Persebaran Kejadian Penting

Tabel 4.3 berikut merupakan tabel use case dari Aplikasi *Incident Report* melihat peta persebaran kejadian penting.

*Tabel 4.3 Tabel Use Case Melihat Peta Persebaran Kejadian Penting*

Nama	Melihat peta persebaran kejadian penting
Kode	UC001
Deskripsi	Aktor dapat melihat peta persebaran kejadian penting
Tipe	Fungsional
Pemicu	1. Aktor membuka aplikasi 2. Aktor menekan menu home
Aktor	Masyarakat Jawa Timur
Kondisi Awal	-
Kondisi Akhir	Aktor dapat melihat peta persebaran kejadian penting

Alur Kejadian Secara Normal	1a. Aktor membuka aplikasi 1b. Aktor menekan menu home 1. Sistem menampilkan peta persebaran kejadian dengan 4 warna titik berbeda, yaitu : <ul style="list-style-type: none"> <li>- Merah : Kebakaran</li> <li>- Kuning : Lalu-lintas</li> <li>- Hijau : Bencana Alam</li> <li>- Abu-abu : Lain-lain</li> </ul> 2. Aktor menekan titik untuk menampilkan tweet kejadian 3. Aktor menekan ikon garis 3 4. Aktor memilih rentang waktu dan menekan submit 5. Sistem menampilkan peta persebaran kejadian penting pada rentang waktu yang dipilih
Alur Kejadian Alternatif	-
Pengecualian	-

#### 4.3.2. Melihat Data *Tweet* Kejadian Penting

Tabel 4.4 berikut merupakan tabel use case dari Aplikasi *Incident Report* melihat data tweet kejadian penting.

*Tabel 4.4 Tabel Use Case Melihat Data Tweet Kejadian Penting*

Nama	Melihat data tweet kejadian penting
Kode	UC002
Deskripsi	Aktor dapat melihat data tweet kejadian penting
Tipe	Fungsional
Pemicu	1. Aktor menekan menu Daftar Kejadian
Aktor	Masyarakat Jawa Timur
Kondisi	-



Awal	
Kondisi Akhir	Aktor dapat melihat data tweet kejadian penting
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> <li>1. Aktor menekan menu Daftar Kejadian</li> <li>2. Sistem menampilkan data tweet kejadian penting</li> </ol>
Alur Kejadian Alternatif	-
Pengecualian	-

#### 4.3.3. Melihat Rekapitulasi Kejadian

Tabel 4.5 berikut merupakan tabel use case dari Aplikasi *Incident Report* melihat rekapitulasi kejadian.

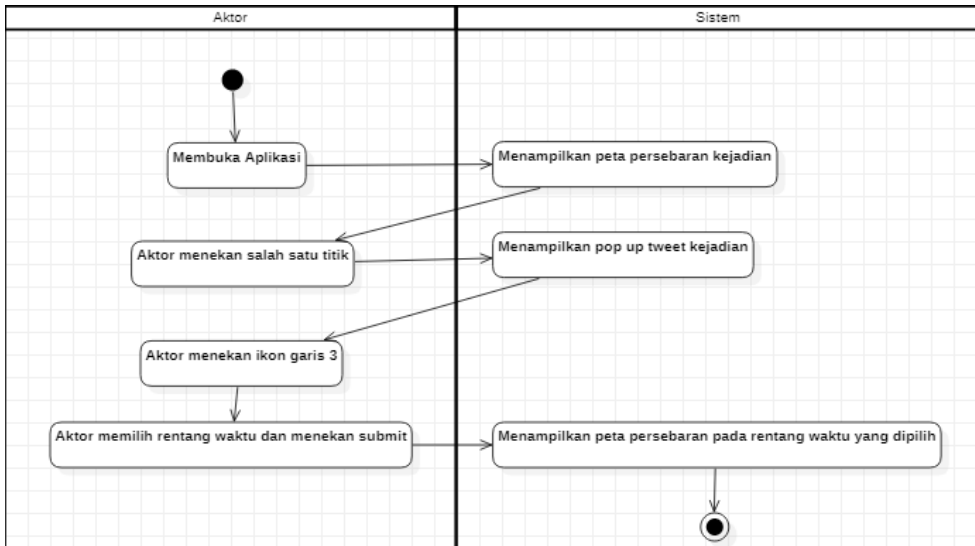
*Tabel 4.5 Tabel Use Case Melihat Rekapitulasi Kejadian*

Nama	Melihat rekapitulasi kejadian
Kode	UC003
Deskripsi	Aktor dapat melihat rekapitulasi kejadian
Tipe	Fungsional
Pemicu	1. Aktor menekan menu Rekapitulasi Kejadian
Aktor	Masyarakat Jawa Timur
Kondisi Awal	-
Kondisi Akhir	Aktor dapat melihat rekapitulasi kejadian
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> <li>1. Aktor menekan menu Rekapitulasi Kejadian</li> <li>2. Sistem menampilkan rekapitulasi kejadian berdasarkan bulan dan kecamatan</li> </ol>
Alur Kejadian Alternatif	-
Pengecualian	-

#### 4.4. Diagram Aktivitas

##### 4.4.1. Melihat Peta Persebaran Kejadian Penting

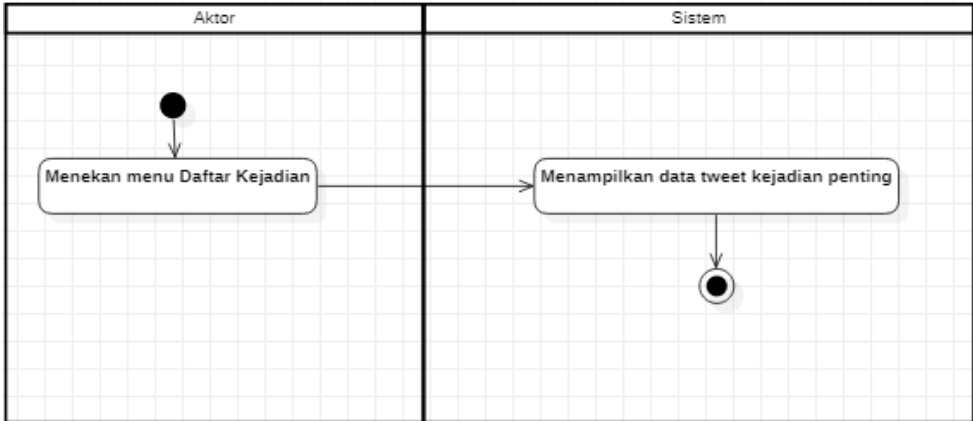
Gambar 4.3 berikut merupakan Diagram Aktivitas dari Aplikasi *Incident Report* untuk melihat peta persebaran kejadian penting.



Gambar 4.3 Diagram Aktivitas Menampilkan Peta Persebaran Kejadian Penting

#### 4.4.2. Melihat Data *Tweet* Kejadian Penting

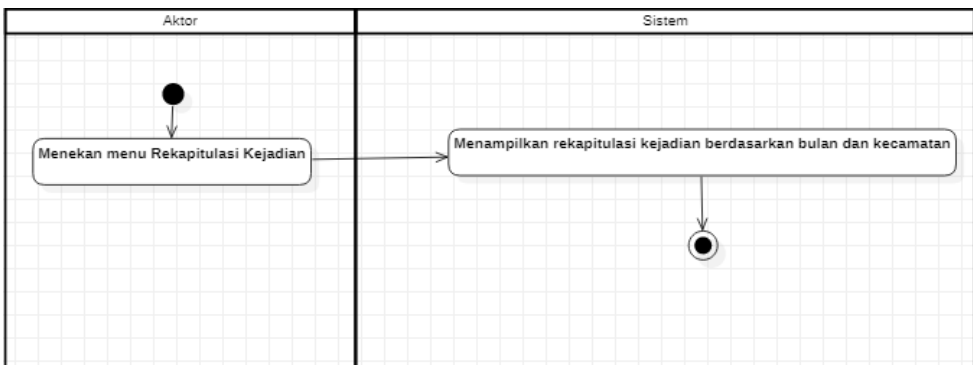
Gambar 4.4 berikut merupakan Diagram Aktivitas dari Aplikasi *Incident Report* untuk melihat data tweet kejadian penting.



Gambar 4.4 Diagram Aktivitas Menampilkan Data Tweet Kejadian Penting

#### 4.4.3. Melihat Rekapitulasi Kejadian

Gambar 4.5 berikut merupakan Diagram Aktivitas dari Aplikasi *Incident Report* untuk melihat rekapitulasi kejadian.



Gambar 4.5 Diagram Aktivitas Menampilkan Rekapitulasi Kejadian

## BAB V

### IMPLEMENTASI SISTEM

Bab ini membahas tentang implementasi dari perancangan sistem dan pengaplikasian sistem dalam bentuk situs web. Implementasi ini terbagi menjadi 2, yaitu implementasi pengambilan data dan implementasi web.

#### 5.1. Implementasi Pengambilan Data

##### 5.1.1. Mengambil Data *Tweet*

Kode sumber 5.1 merupakan pengambilan data *tweet* kejadian penting dari twitter yang berfokuskan pada akun Suara Surabaya.

```
db = redis.Redis(  
    decode_responses=True,  
    host=redis_config['host'],  
    port=redis_config['port']  
)  
logger = logging.getLogger(__name__)  
class CustomListener(StreamListener):  
    def on_status(self, status):  
        raw_tweet = None  
        tweet_id = None  
        user_id = None  
        created_at = None  
        if hasattr(status, 'retweeted_status'):  
            return True  
        else:  
            tweet_id = status.id_str  
            user_id = status.user.id_str  
            created_at = status.created_at  
            try:  
                raw_tweet = status.extended_tweet["full_text"]  
            except:  
                raw_tweet = status.text  
  
        payload = {  
            'text': raw_tweet,  
            'created_at': str(created_at),  
            'tweet_id': tweet_id,  
            'user_id': user_id
```

```

    }
    logger.info(payload)
    processed_text = clean_proc(raw_tweet)
    for text in processed_text:
        payload['processed_text'] = text
        dumps = json.dumps(payload)
        db.lpush(STREAM_LIST_KEY, dumps)
    return True

def on_error(self, status_code):
    if status_code == 420:
        logger.warning('rate limit')
    return False

def back_off(pending):
    if pending > 500:
        pending = 1
    logger.warning('Back off { } seconds'.format(pending))
    time.sleep(pending)
    pending *= 2
    return pending

if __name__ == '__main__':
    log_dir = os.path.join("../storage", "streamer.log")
    c_handler = logging.StreamHandler()
    f_handler = logging.FileHandler(log_dir)

    formatter = logging.Formatter('%(asctime)s - STREAMER - %(levelname)s - %(message)s')
    c_handler.setFormatter(formatter)
    f_handler.setFormatter(formatter)

    logger.addHandler(c_handler)
    logger.addHandler(f_handler)
    logger.setLevel(logging.INFO)
    logger.info("Start")

    pending = 1
    max_pending = 500
    while True:
        try:
            l = CustomListener()
            auth = OAuthHandler(APP_KEY, APP_SECRET)
            auth.set_access_token(OAUTH_TOKEN, OAUTH_TOKEN_SE
CRET)

```

```

stream = Stream(auth, l)
stream.filter(follow=TARGET_USER_ID)
except KeyboardInterrupt:
    break
except Exception:
    logger.error("Exception occur", exc_info=True)
    pending = back_off(pending)
else:
    pending = back_off(pending)

```

*Kode Sumber 5.1 Mengambil Data Tweet*

### 5.1.2. Melakukan *Pre-Processing* Data

Kode Sumber 5.2 merupakan *pre-processing* dari data *tweet* yang telah diambil, dengan langkah-langkah sebagai berikut :

1. Memecah kalimat menjadi kata-kata
2. Mengganti *recognized entities* dengan jenis entitasnya
3. Menyimpan posisi awal kata yang masuk entitas ['LOC', 'BLD', 'NPL', 'GPE', 'HWYMSE']

```

def delete_repeating_char(text: str) -> str:
    n = len(text)
    if n < 2:
        return text
    result = [None] * n
    result[0] = text[0]
    j = 0
    for i in range(1, n):
        if text[i] != result[j]:
            j += 1
            result[j] = text[i]
    result = list(filter(None, result))
    result = ".join(result)
    return result

def process_entities(token_list: list, analysis: dict) -> (list, list):
    index_shift = 0
    location_entities = {}
    for entity in analysis['entities']:
        start = entity['beginOffset'] - index_shift
        stop = entity['endOffset'] - index_shift
        token_list[start: stop] = [entity['type']]
        index_shift = stop - start - 1

```

```

        if entity['type'] in ['LOC', 'BLD', 'GPE', 'NPL', 'HWYMSE']: #, 'TIME', 'DATE']:
            index = str(start)
            location_entities[index] = {
                'text': entity['text'],
                'type': entity['type']
            }
        return token_list, location_entities
def validate_location_entities_position(ner_tweet: str, location_entities: Dict[str, Dict[str, str]]) -> Dict[str, Dict[str, str]]:
    tweet_tokens = ner_tweet.split()
    validated_loc_entities = {}
    last_index = 0
    for index, loc_entity in location_entities.items():
        index = int(index)
        location_type = loc_entity["type"].lower()
        if index < len(tweet_tokens) and tweet_tokens[index] == location_type:
            last_index = index
        else:
            found_index = tweet_tokens.index(location_type, last_index)
            last_index = found_index
            validated_loc_entities[str(last_index)] = loc_entity
    return validated_loc_entities

with open('config.yaml', 'r') as f:
    config = yaml.safe_load(f)

ner_config = config.get('ner')
redis_config = config.get('redis')
redis_key = redis_config.get('key')
port=redis_config['port']
)
if __name__ == "__main__":
    log_dir = os.path.join("../storage", "ner2.log")
    c_handler = logging.StreamHandler()
    f_handler = logging.FileHandler(log_dir)

    formatter = logging.Formatter('%(asctime)s - NER - %(levelname)s - %(message)s')
    c_handler.setFormatter(formatter)
    f_handler.setFormatter(formatter)

    logger.addHandler(c_handler)
    logger.addHandler(f_handler)
    logger.setLevel(logging.INFO)

```

```

logger.info("Start")

while True:
    try:
        data = db.rpop(STREAM_LIST_KEY)
        if data is None:
            time.sleep(SLEEP_DURATION)
            continue
        data = json.loads(data)
        logger.info("Processing %s", data)
        processed_tweet = data["processed_text"]
        if not processed_tweet:
            logger.warning("Empty tweet")
            continue
        data["ner_text"] = []
        data["location_entities"] = []
        tweet_tokens = processed_tweet.split()
        analysis = model.analyze(tweet_tokens)
        tweet_tokens, location_entities = process_entities(tweet_tokens,
analysis)
        if len(location_entities) == 0:
            continue
        tweet_ner = " ".join(tweet_tokens)
        tweet = delete_repeating_char(tweet_ner.lower())
        tweet = re.sub(r"^[a-z\?\-]", " ", tweet)
        tweet = re.sub(r"\s{2,}", " ", tweet)
        location_entities = validate_location_entities_position(tweet, loc
ation_entities)

        data["ner_text"] = tweet
        data["location_entities"] = location_entities
        logger.info("Found entities [%s] and tweet-
ner [%s]", location_entities, tweet)
        data = json.dumps(data)
        db.lpush(NER_LIST_KEY, data)
        print('ner:'+data)

```

*Kode Sumber 5.2 Melakukan Pre-Processing Data*

### 5.1.3. Melakukan Klasifikasi Jenis Kejadian

Kode Sumber 5.3 melakukan klasifikasi menjadi 4 jenis kejadian yaitu lalu-lintas, bencana alam, kebakaran, dan lain-lain. Langkah-langkah klasifikasi jenis kejadian yaitu :



1. Melakukan klasifikasi hanya jika ada entitas lokasi di tweet.
2. Menghapus *consecutive repeated* char dari kata-kata di *tweet* yang telah di-*encode* oleh entitas.
3. Menghapus char selain [a-z ? -].
4. Menghapus *whitespace* berlebih pada *tweet*.
5. Mengklasifikasi *tweet*.
6. Melakukan proses lebih lanjut hanya jika bukan tipe 0.
7. Jika entitas posisi == 1, mencari latitude-longitude dari lokasi kejadian, lalu memasukkan ke database.

```

if __name__ == "__main__":
    log_dir = os.path.join("../storage", "incident-classifier.log")
    c_handler = logging.StreamHandler()
    f_handler = logging.FileHandler(log_dir)

    formatter = logging.Formatter('%(asctime)s - INCIDENT_CLASS -
%(levelname)s - %(message)s')
    c_handler.setFormatter(formatter)
    f_handler.setFormatter(formatter)

    logger.addHandler(c_handler)
    logger.addHandler(f_handler)
    logger.setLevel(logging.INFO)
    logger.info("Start")

    text_proc = TextProc()
    checkpoint_file = tf.train.latest_checkpoint(CHECKPOINT_DIR)
    logger.info(checkpoint_file)
    graph = tf.Graph()
    with graph.as_default():
        config = tf.ConfigProto(
            allow_soft_placement=True,
        )
        session = tf.Session(config=config)
        with session.as_default():
            saver = tf.train.import_meta_graph("{} .meta".format(checkpoint
_file))
            saver.restore(session, checkpoint_file)
            input_text = graph.get_operation_by_name("input_text").outputs
[0]
            dropout_keep_prob = graph.get_operation_by_name("dropout_k
eep_prob").outputs[0]

```

```

predictions = graph.get_operation_by_name("output/logits").out
puts
while True:
    try:
        data = db.rpop(NER_LIST_KEY)
        if data is None:
            time.sleep(SLEEP_DURATION)
            continue
        data = json.loads(data)
        logger.info("Processing %s", data)
        ner_text = data["ner_text"]
        ner_text = [ner_text]
        input_data = text_proc.transform(ner_text)
        preds = session.run(predictions, {input_text: input_data,
                                         dropout_keep_prob: 1.0})

        # softmax prob
        probabilities = softmax(preds[0], axis=1)
        prediction_result = np.argmax(probabilities, axis=1)
        logger.info("Hasil prediksi softmax %s", probabilities)
        logger.info("Hasil argmax %s", np.argmax(prediction_resul
t, axis=0))

        result = prediction_result[0]
        if probabilities[0][result] < MIN_THRESHOLD:
            result = 0
            logger.warning("Prediksi dengan prob %s di bawah thres
hold %s", probabilities[0][result], MIN_THRESHOLD)
        if result == 0:
            logger.info("Bukan kejadian penting")
            continue
        label = CLASS_LABEL[result]
        data["incident_label"] = label
        if len(data["location_entities"]) == 1:
            logger.info("Geocode dengan label %s", label)
            location_entities = list(data["location_entities"].values())
            data["location"] = {
                "source": location_entities[0]["text"]
            }
        payload = json.dumps(data)
        db.lpush(GEOCODE_LIST_KEY, payload)
        continue

```

*Kode Sumber 5.3 Melakukan Klasifikasi Jenis Kejadian*

#### 5.1.4. Melakukan Ekstraksi Kejadian

Kode Sumber 5.4 ekstraksi relasi kejadian dengan langkah sebagai berikut :

1. Membuat semua kombinasi pasangan entitas lokasi di *tweet*.
2. Mengubah menjadi format yang memiliki <e1> dan <e2> , menghapus *multiple whitespace*.
3. Memproses ke *vocab processor* (mengubah menjadi *sequence of word*).
4. Memprediksi relasi.
5. Menentukan *startpoint*, *waypoint*, dan *endpoint* berdasarkan relasi di graf.
6. Memasukkan ke database.

```
def location_entities_combination(location_entities: dict) -
> List[Tuple[int, int]]:
    indexes = []
    for index in location_entities.keys():
        indexes.append(int(index))
    return itertools.combinations(indexes, 2)
def generate_input_data(data: dict) -> List[str]:
    input_data = []
    ner_token = data["ner_text"].split()
    pattern = re.compile(r"\s{2,}")
    for pair in location_entities_combination(data["location_entities"]):
        tokens = ner_token.copy()
        first, second = pair
        tokens[first] = "<e1>{ }<e1>".format(ner_token[first])
        tokens[second] = "<e2>{ }<e2>".format(ner_token[second])
        text = " ".join(tokens)
        text = pattern.sub(" ", text)
        input_data.append(text.lower())
    return input_data
def process_location_entities_graph(
    ner_text_token: List[str],
    location_entities: dict,
    predictions: List[int],
    location_entities_combination: List[Tuple[int, int]]) -
> networkx.DiGraph:
    graph = networkx.DiGraph()
    for loc_entity in location_entities.values():
        graph.add_node(loc_entity["text"])
```

```

relation_predictions, combinations = list(
    zip(*sorted(zip(predictions, location_entities_combination), revers
e=True)))
removed_node = set()
for relation_prediction, combination in zip(relation_predictions, comb
inations):
    first_entity, second_entity = combination
    first_entity_type = ner_text_token[first_entity]
    second_entity_type = ner_text_token[second_entity]

    first_entity_text = location_entities[str(first_entity)]["text"]
    second_entity_text = location_entities[str(second_entity)]["text"]
    if first_entity_type != HIGHWAY and second_entity_type != HIG
HWAY:
        if relation_prediction == 5:
            graph.add_edge(first_entity_text, second_entity_text)
        elif relation_prediction == 6:
            graph.add_edge(second_entity_text, first_entity_text)
    elif relation_prediction == 1: # hwy-mse 1,2
        removed_node.add(second_entity_text)
    elif relation_prediction == 2: # hwy-mse 2,1
        removed_node.add(first_entity_text)
    elif relation_prediction == 3: # street-place 1,2
        removed_node.add(second_entity_text)
        join_node = second_entity_text + ' ' + first_entity_text
        mapping = {first_entity_text: join_node}
        networkx.relabel_nodes(graph, mapping, copy=False)
    elif relation_prediction == 4: # street-place 2,1
        removed_node.add(first_entity_text)
        join_node = first_entity_text + ' ' + second_entity_text
        mapping = {second_entity_text: join_node}
        networkx.relabel_nodes(graph, mapping, copy=False)

for node in removed_node:
    graph.remove_node(node)
return graph
def get_source_ways_dest(graph: networkx.DiGraph) -
> Tuple[str, Optional[List[str]], str]:
    source = None
    destination = None
    waypoints = []
    if len(graph) < 2:
        node = list(graph.nodes())[0]
        source = gazetter.lookup_compound(node, 4)[0].term
    else:

```

```

for node in graph.nodes():
    predecessor = list(graph.predecessors(node))
    successor = list(graph.successors(node))
    location_name = gazetter.lookup_compound(node, 4)[0].term
    if successor and not predecessor:
        source = location_name
    elif predecessor and not successor:
        destination = location_name
    else:
        waypoints.append(location_name)
return source, waypoints, destination
def add_location(data: dict, source: str, ways: list, destination: str) -
> dict:
    location = { }
    if source:
        location["source"] = source
    if ways:
        location["ways"] = ways
    if destination:
        location["destination"] = destination
    data["location"] = location
return data

```

*Kode Sumber 5.4 Melakukan Ekstraksi Kejadian*

### 5.1.5. Menemukan Tempat Kejadian

Kode Sumber 5.5 menentukan tempat kejadian tweet berdasarkan longitude-latitude.

```

def find_place(query):
    name = None
    address = None
    latitude = None
    longitude = None
    place_result = gmaps.find_place(input=query, input_type="textquery",
    fields=['geometry', 'name', 'place_id', 'formatted_address'],
    location_bias="circle:50@-7.279268,112.797217")
    if len(place_result['candidates']) == 0:
        return name, address, latitude, longitude

    name = place_result['candidates'][0]['name']
    address = place_result['candidates'][0]['formatted_address']

```

```

latitude = place_result['candidates'][0]['geometry']['location']['lat']
longitude = place_result['candidates'][0]['geometry']['location']['lng']

return name, address, latitude, longitude

if __name__ == "__main__":
    log_dir = os.path.join("../storage", "geocode.log")
    c_handler = logging.StreamHandler()
    f_handler = logging.FileHandler(log_dir)

    formatter = logging.Formatter('%(asctime)s - GEOCODE - %(levelname)s - %(message)s')
    c_handler.setFormatter(formatter)
    f_handler.setFormatter(formatter)

    logger.addHandler(c_handler)
    logger.addHandler(f_handler)
    logger.setLevel(logging.INFO)
    logger.info("Start")

    while True:
        try:
            data = db.rpop(GEOCODE_LIST_KEY)
            if data is None:
                time.sleep(SLEEP_DURATION)
                continue

            data = json.loads(data)
            logger.info("Processing %s", data)

            new_data = {
                "tweet_id": data["tweet_id"],
                "user_id": data["user_id"],
                "tweet": data["text"],
                "created_at": data["created_at"],
                "incident_type": data["incident_label"]
            }

            if "source" not in data["location"]:
                logger.warning("No source location found. Not processed")
                continue

            if "source" in data["location"]:
                logger.info("Get source")
                loc_source = data["location"]["source"]
                name, address, latitude, longitude = find_place(loc_source)

```

```

new_data["source_name"] = name
new_data["source_address"] = address
new_data["latitude_src"] = latitude
new_data["longitude_src"] = longitude

if "destination" in data["location"]:
    logger.info("Get destination")
    loc_dest = data["location"]["destination"]
    name, address, latitude, longitude = find_place(loc_dest)

    new_data["destination_name"] = name
    new_data["destination_address"] = address
    new_data["latitude_dst"] = latitude
    new_data["longitude_dst"] = longitude

if "ways" in data["location"]:
    logger.info("Get waypoints")
    way_name = []
    way_address = []
    way_latitude = []
    way_longitude = []

    for loc_way in data["location"]["ways"]:
        name, address, latitude, longitude = find_place(loc_way)
        way_name.append(name)
        way_address.append(address)
        way_latitude.append(latitude)
        way_longitude.append(longitude)

```

*Kode Sumber 5.5 Menemukan Tempat Kejadian*

### 5.1.6. Memasukkan Kedalam Database

Kode Sumber 5.6 memasukkan data yang telah diproses kedalam database.

```

def insert_process(cursor, data: tuple, query: str):
    success = False
    for i in range(MAX_RETRY):
        try:
            cursor.execute(query, data)
            connection.commit()
            success = True
            break

```

```

except OperationalError:
    logger.warning("[RETRY] Attempt %s", i)
    connection = mysql.connector.connect(
        host=database_config['host'],
        database=database_config['name'],
        user=database_config['user'],
        password=database_config['pass']
    )
    cursor = connection.cursor()
if not success:
    logger.error("Failed to insert data")
else:
    logger.info("Success")

if __name__ == "__main__":
    log_dir = os.path.join("../storage", "store.log")
    c_handler = logging.StreamHandler()
    f_handler = logging.FileHandler(log_dir)

    formatter = logging.Formatter('%(asctime)s - STORE - %(levelname)s - %(message)s')
    c_handler.setFormatter(formatter)
    f_handler.setFormatter(formatter)

    logger.addHandler(c_handler)
    logger.addHandler(f_handler)
    logger.setLevel(logging.INFO)
    logger.info("Start")

while True:
    try:
        cursor = None
        data = db.rpop(STORE_LIST_KEY)
        if data is None:
            time.sleep(SLEEP_DURATION)
            continue
        data = json.loads(data)
        logger.info("Processing %s", data)
        tweet_id = data["tweet_id"]
        user_id = data["user_id"]
        tweet = data["tweet"]
        date = datetime.strptime(data["created_at"], "%Y-%m-%d %H:%M:%S")
        date = date + timedelta(hours=7)
        created_at = str(date)

```



```

if data.get("source_name") is None:
    logger.warning("Missing source name")
    continue
incident_type = data.get("incident_type")
source_name = data.get("source_name")
source_address = data.get("source_address")
latitude_src = data.get("latitude_src")
longitude_src = data.get("longitude_src")
destination_name = data.get("destination_name")
destination_address = data.get("destination_address")
latitude_dst = data.get("latitude_dst")
longitude_dst = data.get("longitude_dst")
way_name = data.get("way_name")
way_address = data.get("way_address")
latitude_way = data.get("latitude_way")
longitude_way = data.get("longitude_way")
success = False
insert_tuple = (tweet_id, user_id, tweet, created_at, incident_type,
source_name, source_address, latitude_src, longitude_src, destination
_name, destination_address, latitude_dst, longitude_dst, way_name, wa
y_address, latitude_way, longitude_way)
# insert_process(cursor, insert_tuple, sql_insert_query)
for i in range(MAX_RETRY):
    try:
        cursor = connection.cursor()
        cursor.execute(sql_insert_query, insert_tuple)
        connection.commit()
        success = True
        break
    except OperationalError:
        logger.warning("[RETRY] Attempt %s", i)
        connection = mysql.connector.connect(
            host=database_config['host'],
            database=database_config['name'],
            user=database_config['user'],
            password=database_config['pass']
        )
        cursor = connection.cursor()
if not success:
    logger.error("Failed to insert data")
else:
    logger.info("Success")

```

*Kode Sumber 5.6 Memasukkan Kedalam Database*

## 5.2. Implementasi Web

### 5.2.1. Tweet Controller

Berikut Kode Sumber 5.7 php yang digunakan untuk pengambilan detail informasi *tweet*.

```
<?php

namespace App\Http\Controllers;
use App\Tweet;
use Illuminate\Support\Facades\DB;
use Illuminate\Http\Request;
use Carbon\Carbon;

class TweetController extends Controller
{

    public function rekap() {
        if (request()->route()->named('home')){
            $start = Carbon::today();
            $end = Carbon::today()->addDay();
            $tweets = Tweet::whereBetween('created_at', [$start, $end])->get();

            return view('welcome')->with([
                'tweets' => $tweets
            ]);
        }
        elseif (request()->route()->named('daftar-kejadian')){
            $tweets = Tweet::orderBy("created_at", "desc")->get();

            $kecamatan = ["Asemrowo", "Benowo", "Bubutan", "Bulak", "Dukuhpakis", "Gayungan", "Genteng", "Gubeng", "Gn. Anyar", "Jambangan", "Karang Pilang", "Kenjeran", "Krembangan", "Lakarsantri", "Mulyorejo", "Pabean Cantian", "Pakal", "Rungkut", "Sambikerep", "Sawahan", "Semampir", "Simokerto", "Sukolilo", "Sukomanunggal", "Tambaksari", "Tandes", "Tegalsari", "Tenggilis Mejoyo", "Wiyung", "Wonocolo", "Wonokromo"];

            return view('daftar-kejadian')->with([
                'tweets' => $tweets,
                'kecamatan' => $kecamatan
            ]);
        }
        elseif (request()->route()->named('rekap-kejadian')){
```

```

$data = Tweet::selectRaw('count(*) AS cnt, incident_type, MONTH(created_at) as month, YEAR(created_at) as year')
->groupBy('incident_type', 'month', 'year')->get();

$rekap = Array();
foreach($data as $d) {
    $key = (string) $d->month . '-' . (string) $d->year;
    if(array_key_exists($key, $rekap)){
        $rekap[$key][$d->incident_type] = $d->cnt;
    }
    else{
        $rekap[$key] = Array();
        $rekap[$key][$d->incident_type] = $d->cnt;
    }
}

$kecamatan = ["Asemrowo", "Benowo", "Bubutan", "Bulak", "Dukuhpakis", "Gayungan", "Genteng", "Gubeng", "Gn. Anyar", "Jambangan", "Karang Pilang", "Kenjeran", "Krembangan", "Lakarsantri", "Mulyorejo", "Pabean Cantian", "Pakal", "Rungkut", "Sambikerep", "Sawahan", "Semampir", "Simokerto", "Sukolilo", "Sukomanunggal", "Tambaksari", "Tandes", "Tegalsari", "Tenggilis Mejoyo", "Wiyung", "Wonocolo", "Wonokromo"];

$data2 = Tweet::selectRaw('count(*) AS cnt, incident_type, source_address as address, YEAR(created_at) as year')
->groupBy('incident_type', 'address', 'year')->get();
foreach($data2 as $d) {
    $flag = false;
    foreach($kecamatan as $kec) {
        $kecamatanFull = 'Kec. ' . $kec;
        // echo $kecamatanFull . " " . $d->address . "<br>";
        if(strpos($d->address, $kecamatanFull) !== false) {
            $flag = true;
            $key = (string) $kec . '-' . (string) $d->year;
            if(array_key_exists($key, $rekap)){
                if(array_key_exists($d->incident_type, $rekap[$key])) {
                    $rekap[$key][$d->incident_type] += $d->cnt;
                }
                else {
                    $rekap[$key][$d->incident_type] = $d->cnt;
                }
            }
            else{
                $rekap[$key] = Array();
            }
        }
    }
}

```

```

        if(array_key_exists($d-
>incident_type, $rekap[$key])) {
            $rekap[$key][$d->incident_type] += $d->cnt;
        }
        else {
            $rekap[$key][$d->incident_type] = $d->cnt;
        }
    }
    break;
}
}
if($flag == false) {
    $key = 'Lain lain-' . (string) $d->year;
    if(array_key_exists($key, $rekap)){
        if(array_key_exists($d->incident_type, $rekap[$key])) {
            $rekap[$key][$d->incident_type] += $d->cnt;
        }
        else {
            $rekap[$key][$d->incident_type] = $d->cnt;
        }
    }
    else{
        $rekap[$key] = Array();
        if(array_key_exists($d->incident_type, $rekap[$key])) {
            $rekap[$key][$d->incident_type] += $d->cnt;
        }
        else {
            $rekap[$key][$d->incident_type] = $d->cnt;
        }
    }
}
}

return view('rekap-kejadian')->with([
    'rekap' => $rekap
]);
}

}

public function filter_map(Request $request) {
    $timestart = $request->input('timestart');
    $timeend = $request->input('timeend');

    $startcarbon = Carbon::parse($timestart)->format('Y-m-d H:i:s');
    $endcarbon = Carbon::parse($timeend)->format('Y-m-d H:i:s');

```

```

    $tweets = Tweet::whereBetween('created_at', [$startcarbon, $endcarbon])
        ->where(function ($query) {
            $query->whereNull('destination_address')
            -
        })
    >orWhere('destination_address', 'like', '%Jawa Timur%');
    })
    ->where('source_address', 'like', '%Jawa Timur%')
    ->orderBy("created_at", "desc")
    ->get();

    return view('welcome')->with([
        'tweets' => $tweets
    ]);
}

public function filter_rekap(Request $request) {
    $timestart = $request->input('timestart');
    $timeend = $request->input('timeend');

    $startcarbon = Carbon::parse($timestart)->format('Y-m-d H:i:s');
    $endcarbon = Carbon::parse($timeend)->format('Y-m-d H:i:s');

    $kecamatan = ["Asemrowo", "Benowo", "Bubutan", "Bulak", "Dukuhpakis", "Gayungan", "Genteng", "Gubeng", "Gn. Anyar", "Jambangan", "Karang Pilang", "Kenjeran", "Krembangan", "Lakarsantri", "Mulyorejo", "Pabean Cantian", "Pakal", "Rungkut", "Sambikerep", "Sawahan", "Semampir", "Simokerto", "Sukolilo", "Sukomanunggal", "Tambaksari", "Tandes", "Tegalsari", "Tenggilis Mejoyo", "Wiyung", "Wonocolo", "Wonokromo"];
}

```

*Kode Sumber 5.7 Tweet Controller*

### 5.2.2. Daftar Kejadian

Berikut Kode Sumber 5.8 php yang digunakan untuk menampilkan rekapian *tweet* kejadian.

```
@section('body')
<div id="mySidebar" class="sidebar">
  <form action="{ {route('filter-rekap')} }" method="POST">
    {{ csrf_field() }}
    <div class="form-group">
      <p>Rentang Waktu</p>
      <label for="timestart">Dari</label>
      <input type="text" class="form-
control" name="timestart" id="timestart" placeholder="dari"/>
      <label for="timeend">Sampai</label>
      <input type="text" class="form-
control" name="timeend" id="timeend" placeholder="sampai"/>
    </div>
    <button type="submit" class="btn btn-light btn-
block" id="submitfilter">Submit</button>
  </form>

</div>

<div class="container rekap-container">
  <table id="rekap-table" class="table table-striped table-
bordered" style="width:100%; display:none;">
    <thead>
      <tr>
        <th>ID</th>
        <th>User ID</th>
        <th>Tweet</th>
        <th>Created at</th>
        <th>Incident type</th>
        <th>Source name</th>
        <th>Destination name</th>
        <th>Way name</th>
        <th>Source address</th>
        <th>Destination address</th>
        <th>Way address</th>
        <th>LatLng source</th>
        <th>LatLng destination</th>
        <th>LatLng way</th>
        <th>Kecamatan</th>
      </tr>
    </thead>
```

```

<tbody>
  @foreach ($tweets as $tweet)
    <tr>
      <td>{{ $tweet->id }}</td>
      <td>{{ $tweet->user_id }}</td>
      <td>{{ $tweet->tweet }}</td>
      <td>{{ $tweet->created_at }}</td>
      <td>{{ $tweet->incident_type }}</td>
      <td>{{ $tweet->source_name }}</td>
      @if (is_null( $tweet->destination_name ))
        <td>-</td>
      @else
        <td>{{ $tweet->destination_name }}</td>
      @endif

      @if (is_null( $tweet->way_name ))
        <td>-</td>
      @else
        @php
          $ways = json_decode($tweet->way_name);
        @endphp

        <td>
          @php
            $n = count($ways);
            echo "<ul>";
            for ($i = 0; $i < $n; $i++) {
              echo "<li>". $ways[$i]. "</li>";
            }
            echo "</ul>"
          @endphp
        </td>
      @endif

      <td>{{ $tweet->source_address }}</td>

      @if (is_null( $tweet->destination_address ))
        <td>-</td>
      @else
        <td>{{ $tweet->destination_address }}</td>
      @endif

      @if (is_null( $tweet->way_address ))
        <td>-</td>
      @else
        @php

```

```

$ways = json_decode($tweet->way_address);
@endphp

<td>
@php
    $n = count($ways);
    echo "<ul>";
    for ($i = 0; $i < $n; $i++) {
        echo "<li>".$ways[$i]."</li>";
    }
    echo "</ul>"
@endphp
</td>
@endif

<td>{{ $tweet->latitude_src }}, {{ $tweet-
>longitude_src }}</td>

@if (is_null( $tweet->latitude_dst ))
<td>-</td>
@else
<td>{{ $tweet->latitude_dst }}, {{ $tweet-
>longitude_dst }}</td>
@endif

@if (is_null( $tweet->latitude_way ))
<td>-</td>
@else
@php
    $lat_ways = json_decode($tweet->latitude_way);
    $lng_ways = json_decode($tweet->longitude_way);
@endphp

<td>
@php
    $n = count($lat_ways);
    echo "<ul>";
    for ($i = 0; $i < $n; $i++) {
        echo "<li>".substr($lat_ways[$i], 0, 9).", ".substr(
$lng_ways[$i], 0, 9)."</li>";
    }
    echo "</ul>"
@endphp
</td>

@endif

```



```

        @php
            $flag = 0;
            foreach($kecamatan as $kec) {
                $kecamatanFull = "Kec. " . $kec;
                if(strpos($tweet-
>source_address, $kecamatanFull)) {
                    echo "<td>" . $kec . "</td>";
                    $flag = 1;
                    break;
                }
            }
            if($flag == 0) {
                echo "<td>Lain lain</td>";
            }
        @endphp
    </tr>
@endforeach

</tbody>
</table>
</div>

@endsection

@section('js')
    <script>
        $(document).ready(function(){
            $(".navbar .navbar-nav .nav-item:nth-
child(2)").addClass("active");

            $('#rekap-table').DataTable({
                scrollX: true,
                "order": [],
            });
            $('#table').show();
            window.dispatchEvent(new Event('resize'));

            $(function() {
                $('#input[name="timestart"]').daterangepicker({
                    singleDatePicker: true,
                    showDropdowns: true,
                    minYear: 2010,
                    maxYear: parseInt(moment().format('YYYY'),10),
                    timePicker: true,
                    timePicker24Hour: true,
                    locale: {

```

```

        format: 'YYYY-MM-DD HH:mm:ss'
    }
});

$(input[name="timeend"]).daterangepicker({
    singleDatePicker: true,
    showDropdowns: true,
    minYear: 2010,
    maxYear: parseInt(moment().format('YYYY'),10),
    timePicker: true,
    timePicker24Hour: true,
    locale: {
        format: 'YYYY-MM-DD HH:mm:ss'
    }
});

function openNav() {
    document.getElementById("mySidebar").style.width = "250px";
    $(this).one("click", closeNav);
}
function closeNav() {
    document.getElementById("mySidebar").style.width = "0";
    $(this).one("click", openNav);
}

$("#togglesidebarbtn").one("click", openNav);

```

*Kode Sumber 5.8 Daftar Kejadian*

### 5.2.3. Grafik Kejadian

Berikut Kode Sumber 5.9 php yang digunakan untuk menampilkan grafik kejadian berdasarkan kategori bulan dan kecamatan dalam kurun tahun tertentu.

```

<div class="container rekap-container">
    <div class="ml-auto mr-auto" style="width: 10%;">
        <select name="year" class="form-control" id="dropdownYear">
        </select>
    </div>
    <div class="row my-2">
        <div class="col-md-12">
            <div class="card">

```

```

        <div class="card-body">
            <canvas id="chBar" height="100"></canvas>
        </div>
    </div>
</div>

<div class="row my-2">
    <div class="col-md-12">
        <div class="card">
            <div class="card-body">
                <canvas id="chBar2" height="100"></canvas>
            </div>
        </div>
    </div>
</div>
</div>
</div>

@endsection

@section('js')
    <script>
        $(document).ready(function(){
            $(".navbar .navbar-nav .nav-item:nth-
child(3)").addClass("active");
        });

        $('#dropdownYear').each(function() {

            var year = (new Date()).getFullYear();
            var current = year;
            year = 2019;
            for (var i = year; i <= current; i++) {
                if (i == current) {
                    $(this).append('<option selected value="' + i + '">' + i + '</o
ption>');
                }
                else {
                    $(this).append('<option value="' + i + '">' + i + '</option>');
                }
            }
        })

        var year = document.getElementById('dropdownYear').value;

```

```

function rekap_diagram($year) {
    var month_dic = ["Januari", "Februari", "Maret", "April", "Mei",
"Juni",
        "Juli", "Agustus", "September", "Oktober", "Novembe
r", "Desember"];
    var labelsX = [];
    var laluLintas = [];
    var kebakaran = [];
    var bencanaAlam = [];
    var lainlain = [];

    for(var d in month_dic){
        label = month_dic[d] + " " + String(year);
        labelsX.push(label);

        var d2 = parseInt(d) + 1;
        var d3 = String(d2) + "-" + String(year);
        if(!Object.keys(rekap).includes(d3)) {
            laluLintas.push(0);
            kebakaran.push(0);
            bencanaAlam.push(0);
            lainlain.push(0);
            continue;
        }
        laluLintas.push(rekap[d3]["Lalu-Lintas"]);
        kebakaran.push(rekap[d3]["Kebakaran"]);
        bencanaAlam.push(rekap[d3]["Bencana-Alam"]);
        lainlain.push(rekap[d3]["Lain-Lain"]);
    }

    var colors = ['#F4D03F', '#E74C3C', '#52BE80', '#d3d3d3'];

    var chBar = document.getElementById("chBar");

    var chartData = {
        labels: labelsX,
        datasets: [{
            label: "Lalu-Lintas",
            data: laluLintas,
            backgroundColor: colors[0]
        },
        {
            label: "Kebakaran",
            data: kebakaran,
            backgroundColor: colors[1]
        },
    ]
}

```

```

        {
            label: "Bencana-Alam",
            data: bencanaAlam,
            backgroundColor: colors[2]
        },
        {
            label: "Lain-Lain",
            data: lainlain,
            backgroundColor: colors[3]
        }
    ]
};

if (chBar) {
    new Chart(chBar, {
        type: 'bar',
        data: chartData,
        options: {
            scales: {
                xAxes: [{
                    barPercentage: 0.4,
                    categoryPercentage: 0.5
                }],
                yAxes: [{
                    ticks: {
                        beginAtZero: false
                    }
                }]
            },
            title: {
                display: true,
                text: "Diagram Rekapitulasi Kejadian",
                fontSize: 18
            },
            legend: {
                position: 'right'
            },
            responsive: true,
        }
    });
}

function rekap_kec($year) {

```

```

var kec_dic = ["Asemrowo", "Benowo", "Bubutan", "Bulak", "D
ukuhpakis", "Gayungan", "Genteng", "Gubeng", "Gn. Anyar", "Jambang
an", "Karang Pilang", "Kenjeran", "Krembangan", "Lakarsantri", "Muly
orejo", "Pabean Cantian", "Pakal", "Rungkut", "Sambikerep", "Sawahan
", "Semampir", "Simokerto", "Sukolilo", "Sukomanunggal", "Tambaksa
ri", "Tandes", "Tegalsari", "Tenggilis Mejoyo", "Wiyung", "Wonocolo",
"Wonokromo", "Lain lain"];

var labelsX = [];
var laluLintas = [];
var kebakaran = [];
var bencanaAlam = [];
var lainlain = [];

for(var d in kec_dic){
    label = kec_dic[d];
    labelsX.push(label);

    var d2 = String(kec_dic[d]) + "-" + String(year);

    if(!Object.keys(rekap).includes(d2)) {
        laluLintas.push(0);
        kebakaran.push(0);
        bencanaAlam.push(0);
        lainlain.push(0);
        continue;
    }
    laluLintas.push(rekap[d2]["Lalu-Lintas"]);
    kebakaran.push(rekap[d2]["Kebakaran"]);
    bencanaAlam.push(rekap[d2]["Bencana-Alam"]);
    lainlain.push(rekap[d2]["Lain-Lain"]);
}

var colors = ['#F4D03F', '#E74C3C', '#52BE80', '#d3d3d3'];

var chBar = document.getElementById("chBar2");

var chartData = {
    labels: labelsX,
    datasets: [{
        label: "Lalu-Lintas",
        data: laluLintas,
        backgroundColor: colors[0]
    },
    {
        label: "Kebakaran",
        data: kebakaran,

```

```

        backgroundColor: colors[1]
    },
    {
        label: "Bencana-Alam",
        data: bencanaAlam,
        backgroundColor: colors[2]
    },
    {
        label: "Lain-Lain",
        data: lainlain,
        backgroundColor: colors[3]
    }
]
};

if (chBar) {
    new Chart(chBar, {
        type: 'bar',
        data: chartData,
        options: {
            scales: {
                xAxes: [{
                    barPercentage: 0.4,
                    categoryPercentage: 0.5
                }],
                yAxes: [{
                    ticks: {
                        beginAtZero: false
                    }
                }]
            },
            title: {
                display: true,
                text: "Diagram Rekapitulasi Kejadian",
                fontSize: 18
            },
            legend: {
                position: 'right'
            },
            responsive: true,
        }
    });
}
}

```

```

var rekap = {!! json_encode($rekap) !!};

rekap_diagram(year);
rekap_kec(year);

$('select').on('change', function() {
    year = this.value;
    rekap_diagram(year);
    rekap_kec(year);
});

</script>

```

*Kode Sumber 5.9 Daftar Kejadian*

#### 5.2.4. Tampilan Web

Berikut Kode Sumber 5.10 javascript yang digunakan sebagai tampilan web *incident report*.

```

function getMarkerImage(incidentType) {
    if(incidentType == "Lalu-Lintas"){
        icon = {
            url: "https://maps.google.com/mapfiles/ms/icons/yellow.png",
            size: new google.maps.Size(32, 32),
            origin: new google.maps.Point(0, 0),
            anchor: new google.maps.Point(16, 32),
            labelOrigin: new google.maps.Point(16, 10)
        }
    }
    else if(incidentType == "Kebakaran"){
        icon = {
            url: "https://maps.google.com/mapfiles/ms/icons/red.png",
            size: new google.maps.Size(32, 32),
            origin: new google.maps.Point(0, 0),
            anchor: new google.maps.Point(16, 32),
            labelOrigin: new google.maps.Point(16, 10)
        }
    }
    else if(incidentType == "Bencana-Alam"){
        icon = {
            url: "https://maps.google.com/mapfiles/ms/icons/green.png",
            size: new google.maps.Size(32, 32),
            origin: new google.maps.Point(0, 0),
            anchor: new google.maps.Point(16, 32),

```



```

        labelOrigin: new google.maps.Point(16, 10)
    }
}
else if(incidentType == "Lain-Lain") {
    icon = {
        url: "https://maps.google.com/mapfiles/ms/icons/grey.png",
        size: new google.maps.Size(32, 32),
        origin: new google.maps.Point(0, 0),
        anchor: new google.maps.Point(16, 32),
        labelOrigin: new google.maps.Point(16, 10)
    }
}

return icon;
}

function getDirectionColor(incidentType) {
    if(incidentType == "Lalu-Lintas"){
        return "yellow";
    }
    else if(incidentType == "Kebakaran"){
        return "red";
    }
    else if(incidentType == "Bencana-Alam"){
        return "green";
    }
    else if(incidentType == "Lain-Lain"){
        return "grey";
    }
}
}

```

*Kode Sumber 5.10 Tampilan Web*

## **BAB VI**

### **PENGUJIAN DAN EVALUASI**

Pengujian dilakukan untuk memastikan kualitas perangkat lunak yang dikembangkan dan kesesuaian hasil eksekusi perangkat lunak dengan analisis dan perancangan perangkat lunak.

#### **6.1. Tujuan Pengujian**

Uji coba dilakukan terhadap pengambilan data dan web untuk menguji kebenaran dan ketepatan fungsionalitas keseluruhan dari modul pengambilan data dan web.

#### **6.2. Kriteria Pengujian**

Penilaian atas pencapaian tujuan pengujian didapatkan dengan memperhatikan beberapa hal berikut:

1. Kemampuan Pengambilan Data dalam mengambil data hanya kejadian penting.
2. Kemampuan Pengambilan Data dalam menghapus data bukan kejadian penting.
3. Kemampuan Web dalam menampilkan rekapitulasi kejadian berdasarkan kategori bulan dan kecamatan.
4. Kemampuan Web dalam menetapkan lokasi kejadian pada *maps*.
5. Kemampuan Web dalam menampilkan data *tweet* kejadian.

#### **6.3. Lingkungan Pengujian**

Lingkungan pengujian sistem pada pengerjaan kerja praktik ini dilakukan pada lingkungan dan alat kakas sebagai berikut:

Processor	: AMD A8-7410 APU with AMD Radeon R5 Graphics
Memory	: 8.00 GB
Jenis Device	: Laptop
Sistem Operasi	: Windows 10 Pro

#### 6.4. Skenario Pengujian

Skenario pengujian dilakukan dengan melakukan peran sebagai user yang akan menjalankan fitur-fitur dan seluruh kebutuhan fungsional dari sistem. Langkah-langkah untuk setiap kebutuhan fungsional yaitu sebagai berikut:

1. User dapat membuka web
2. User dapat melihat *maps* berisikan titik kejadian
3. User dapat membuka salah satu titik
4. User dapat melihat *tweet* dari titik kejadian
5. User dapat memilih rentang waktu kejadian
6. User dapat melihat *maps* berdasarkan rentang waktu yang dipilih
7. User dapat membuka menu daftar kejadian
8. User dapat melihat daftar yang berisikan *tweet* kejadian
9. User dapat membuka menu rekapitulasi kejadian
10. User dapat melihat grafik kejadian berdasarkan bulan
11. User dapat melihat grafik kejadian berdasarkan kecamatan
12. User dapat merubah tahun kejadian
13. User dapat melihat grafik kejadian berdasarkan tahun yang dipilih

#### 6.5. Evaluasi Pengujian

Hasil pengujian dilakukan terhadap pengamatan mengenai perilaku sistem Aplikasi *Incident Report* terhadap kasus skenario uji coba. Tabel 6.1 menjelaskan hasil uji coba terhadap aplikasi yang telah dibuat.

No	Kriteria Pengujian	Hasil Pengujian
1.	Kemampuan Pengambilan Data dalam mengambil data hanya kejadian penting.	Terpenuhi
2.	Kemampuan Pengambilan Data dalam menghapus data bukan kejadian penting.	Terpenuhi
3.	Kemampuan Web dalam menampilkan rekapitulasi kejadian berdasarkan kategori	Terpenuhi

	bulan dan kecamatan.	
4.	Kemampuan Web dalam menetapkan lokasi kejadian pada <i>maps</i> .	Belum Terpenuhi
5.	Kemampuan Web dalam menampilkan data <i>tweet</i> kejadian.	Terpenuhi

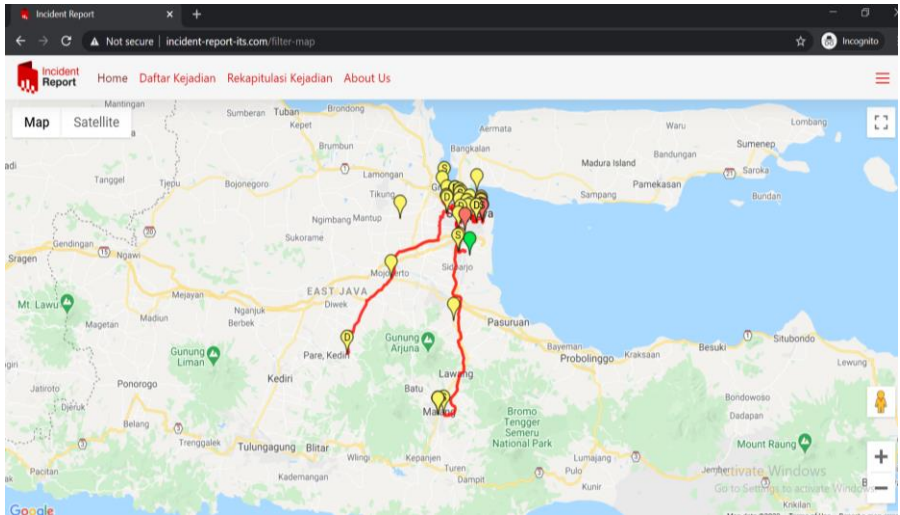
*Tabel 6.1 Hasil Evaluasi Pengujian Aplikasi Incident Report*

Dengan hasil pengujian yang telah ditunjukkan, terdapat satu kriteria yang belum terpenuhi yaitu Kemampuan Web dalam menetapkan lokasi kejadian pada *maps*, dikarenakan terdapat *bug* yaitu nama daerah yang hampir sama sehingga pengambilan longitude-latitude yang salah. Dapat disimpulkan bahwa secara keseluruhan Aplikasi *Incident Report* telah memenuhi kriteria-kriteria yang sudah disebutkan pada bagian-bagian sebelumnya sehingga mampu melewati tahap pengujian yang telah dilakukan.

## 6.6. Tampilan Aplikasi

### 6.6.1. Tampilan Lokasi Kejadian Penting

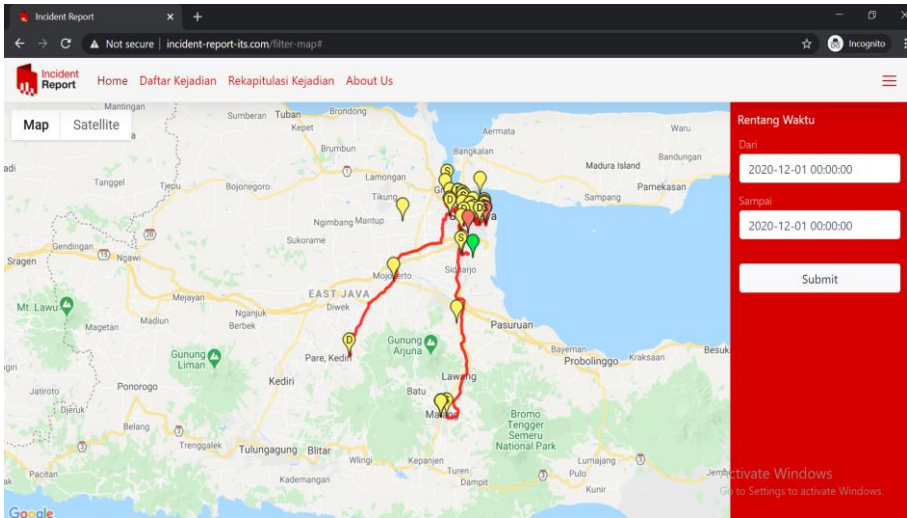
Aplikasi *Incident Report* dengan tampilan lokasi kejadian penting secara visual melalui peta pada Gambar 6.1.



Gambar 6.1 Tampilan Lokasi Kejadian Penting

### 6.6.2. Tampilan Pilihan Rentang Waktu Kejadian Penting

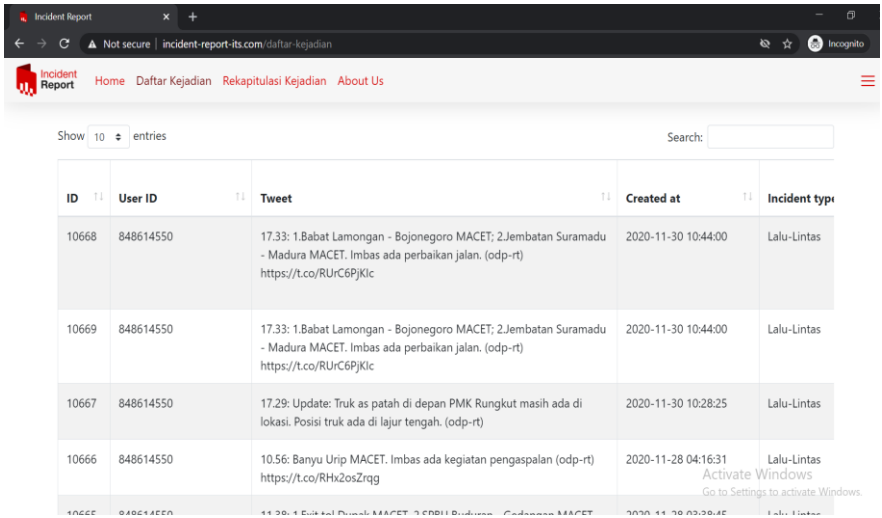
Aplikasi *Incident Report* dengan tampilan pilihan rentang waktu kejadian penting pada Gambar 6.2.



*Gambar 6.2 Tampilan Pilihan Rentang Waktu Kejadian Penting*

### 6.6.3. Tampilan Data Tweet Kejadian Penting

Aplikasi *Incident Report* dengan tampilan data tweet kejadian penting pada Gambar 6.3.



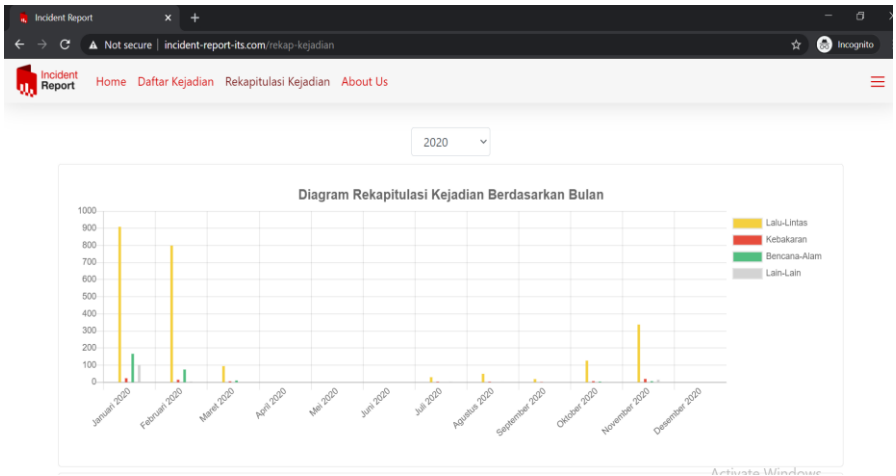
The screenshot shows a web browser displaying the 'Incident Report' application. The page has a navigation bar with links: Home, Daftar Kejadian, Rekapitulasi Kejadian, and About Us. Below the navigation bar, there is a search bar and a table of incident data. The table has five columns: ID, User ID, Tweet, Created at, and Incident type. The table contains five rows of data, each representing a tweet incident. The first four rows are visible, and the fifth row is partially obscured by a watermark.

ID	User ID	Tweet	Created at	Incident type
10668	848614550	17.33: 1.Babat Lamongan - Bojonegoro MACET; 2.Jembatan Suramadu - Madura MACET. Imbas ada perbaikan jalan. (odp-rt) <a href="https://t.co/RUc6PjKlc">https://t.co/RUc6PjKlc</a>	2020-11-30 10:44:00	Lalu-Lintas
10669	848614550	17.33: 1.Babat Lamongan - Bojonegoro MACET; 2.Jembatan Suramadu - Madura MACET. Imbas ada perbaikan jalan. (odp-rt) <a href="https://t.co/RUc6PjKlc">https://t.co/RUc6PjKlc</a>	2020-11-30 10:44:00	Lalu-Lintas
10667	848614550	17.29: Update: Truk as patah di depan PMK Rungkut masih ada di lokasi. Posisi truk ada di lajur tengah. (odp-rt)	2020-11-30 10:28:25	Lalu-Lintas
10666	848614550	10.56: Banyu Urip MACET. Imbas ada kegiatan pengaspalan (odp-rt) <a href="https://t.co/RHx2osZrqg">https://t.co/RHx2osZrqg</a>	2020-11-28 04:16:31	Lalu-Lintas

Gambar 6.3 Tampilan Data Tweet Kejadian Penting

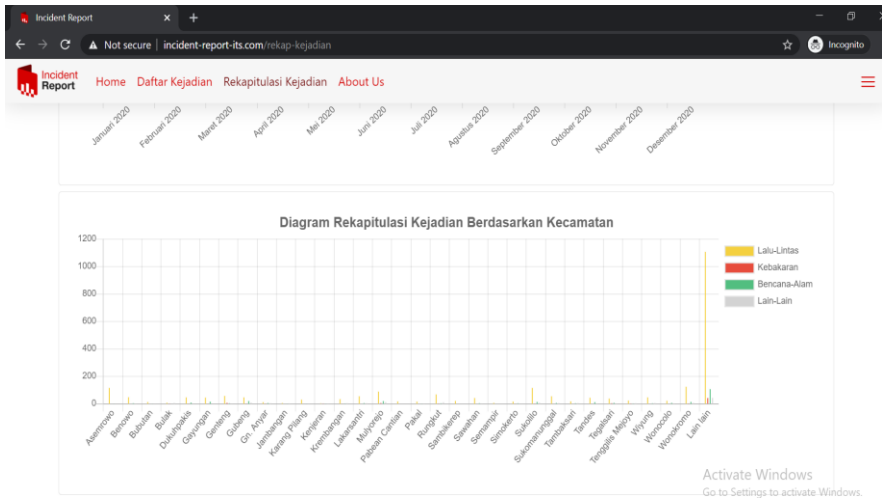
#### 6.6.4. Tampilan Rekapitulasi Kejadian Penting

Aplikasi *Incident Report* dengan tampilan rekapitulasi kejadian penting berdasarkan bulan pada Gambar 6.4 dan tampilan rekapitulasi kejadian penting berdasarkan kecamatan Kota Surabaya pada Gambar 6.5.



Gambar 6.4 Tampilan Rekapitulasi Kejadian Penting Berdasarkan Bulan





*Gambar 6.5 Tampilan Rekapitulasi Kejadian Penting Berdasarkan Kecamatan Kota Surabaya*

## **BAB VII**

### **KESIMPULAN**

#### **7.1. Kesimpulan**

Kesimpulan yang didapat setelah melakukan pengembangan Aplikasi *Incident Report* adalah sebagai berikut:

- Aplikasi yang dibangun telah sesuai dengan permintaan dan dapat dengan mudah dioperasikan oleh pengguna.
- Dengan adanya Aplikasi *Incident Report* dapat memudahkan pengguna, terutama masyarakat Jawa Timur, dalam mendapatkan informasi tentang lalu-lintas, bencana alam, kebakaran dan lain-lain secara *real time*.

#### **7.2. Saran**

Berikut ini adalah beberapa saran yang penulis berikan untuk arah perkembangan selanjutnya:

- Membuat dan atau merubah dokumen standar simulasi aplikasi sesuai dengan penambahan fitur-fitur baru.
- Sangat disarankan untuk memperbaiki *bug* dalam pengambilan longitude-latitude *tweet* untuk menentukan titik lokasi pada *maps*.

## DAFTAR PUSTAKA

- [1] Departemen Teknik Informatika, 2020. *Tentang Informatika*. [online] Available at: <https://www.its.ac.id/informatika/tentang-kami/> [Accessed at 4 November 2020].
- [2] Laboratorium Komputasi Cerdas dan Visi, 2020. *Laboratorium Komputasi Cerdas dan Visi*. [online] Available at: <https://www.its.ac.id/informatika/fasilitas/laboratorium/laboratorium-komputasi-cerdas-dan-visi/> [Accessed at 4 November 2020].
- [3] Janner Simarmata, 2010. *Rekayasa Web*. [online] Available at: <https://books.google.co.id/books?id=J8JpLoPUHGAC&printec=frontcover#v=onepage&q&f=false> [Accessed at 3 November 2020].
- [4] Mozilla Developer Network, 2019. *Dasar Javascript*. [online] Available at: [https://developer.mozilla.org/id/docs/Learn/Getting\\_started\\_with\\_the\\_web/JavaScript\\_basics](https://developer.mozilla.org/id/docs/Learn/Getting_started_with_the_web/JavaScript_basics) [Accessed at 3 November 2020].
- [5] Yunita T., Muhajir A., 2017. *Analisis dan Perancangan Website Sebagai Sarana Informasi Pada Lembaga Bahasa Kewirausahaan Dan Komputer AKMI BATURAJA Menggunakan PHP dan MySQL*. [online] Available at: [journal.binadarma.ac.id/index.php/jurnalmatrik/article/view/366](http://journal.binadarma.ac.id/index.php/jurnalmatrik/article/view/366) [Accessed at 3 November 2020]
- [6] Rinanza Zulmy A., 2016. *Aplikasi Data Pokok Pendidikan Kota Kediri dengan Mengimplementasi CSS Bootstrap*. [online] Available at: [https://www.academia.edu/34876969/APLIKASI\\_DATA\\_PO\\_KOK\\_PENDIDIKAN\\_KOTA\\_KEDIRI\\_DENGAN\\_MENGI\\_MPLEMENTASIAN\\_CSS\\_BOOSTRAP](https://www.academia.edu/34876969/APLIKASI_DATA_PO_KOK_PENDIDIKAN_KOTA_KEDIRI_DENGAN_MENGI_MPLEMENTASIAN_CSS_BOOSTRAP) [Accessed at 3 November 2020]

- [7] Ach Abdul R., 2017. *Aplikasi Penjualan Alat Tulis Kantor (ATK) Berbasis Web Mobile JQuery pada Estu Gresik*. [online] Available at: <http://www.jurnalteknik.unisla.ac.id/index.php/JTIIES/article/view/103> [Accessed at 3 November 2020]
- [8] Id Cloud Host, 2020. *Mengenal Apa Itu Bahasa Pemrograman Python Dan Cara Belajarnya*. [online] Available at : <https://idcloudhost.com/mengenal-apa-itu-bahasa-pemrograman-python-dan-cara-belajarnya/> [Accessed at 3 November 2020]
- [9] Farizan Luthfi, 2017. *Penggunaan Framework Laravel dalam Rancang Bangun Modul Back-End Artikel Website Bisnisnis.ID*. [online] Available at: <http://202.0.92.5/saintek/JISKA/article/view/1165> [Accessed at 3 November 2020]
- [10] Muhamad M., Anggun L., 2014. *Perancangan Sistem Aplikasi Penerimaan Mahasiswa Baru di STMIK Pringsewu Menggunakan PHP dan MySQL*. [online] Available at: <http://ojs.stmikpringsewu.ac.id/index.php/JurnalTam/article/view/24/24> [Accessed at 3 November 2020]
- [11] Rian Yulianto, 2019. *Ekstensi dan Tema Visual Studio Code yang Saya Gunakan*. [online] Available at: <https://medium.com/kode-dan-kodean/ekstensi-dan-tema-visual-studio-code-yang-saya-gunakan-6c3555762816> [Accessed at 3 November 2020]
- [12] Wayan Dadang, 2018. *Memahami Kecerdasan Buatan berupa Deep Learning dan Machine Learning*. [online] Available at: <https://warstek.com/deepmachinelearning/> [Accessed at 11 November 2020]
- [13] Wikipedia, 2020. *Information Extraction*. [online] Available at: [https://en.wikipedia.org/wiki/Information\\_extraction](https://en.wikipedia.org/wiki/Information_extraction) [Accessed at 11 November 2020]
- [14] Franck Dernoncourt, 2020. *NeuroNER*. [online] Available at: <http://neuroner.com/> [Accessed at 11 November 2020]
- [15] P. Gupta, C. Technology, B. Andrassy, and C. Techonology, 2016. *Table Filling Multi-Task Recurrent Neural Network for Joint Entity and Relation Extraction*. pp. 2537-2547

- [16] S. Lai, L. Xu, K. Liu, and J. Zhao, 2015. *Recurrent Convolutional Neural Network for Text Classification*. pp. 2267-2273.
- [17] W. Garbe, 2015. *Fast Approximate String Matching with Large Edit Distances in Big Data*. [online] Available at: <http://blog.faroo.com> [Accessed at 11 November 2020]
- [18] Prio Handoko, 2020. *Struktur Data dan Algoritma*. [online] Available at: <http://www.ocw.upj.ac.id/files/Slide-IFA106-IFA106-Slide-13.pdf> [Accessed at 14 November 2020]

## BIODATA PENULIS



Sherly Rosa Anggraeni, lahir pada tanggal 10 Juli 2000 di Malang. Sebagai mahasiswa yang sedang menempuh studi di Departemen Informatika, Institut Teknologi Sepuluh Nopember (ITS).

Timothyus Tanner, lahir pada tanggal 10 Januari 1998 di Mojokerto. Sebagai mahasiswa yang sedang menempuh studi di Departemen Informatika, Institut Teknologi Sepuluh Nopember (ITS).

